Record Linkage: Theory and Practice

William E. Yancey
william.e.yancey@census.gov

U.S. Census Bureau
Introduction

Definition
Introduction

- Definition
- Terminology
Introduction

- Definition
- Terminology
- Uses
Introduction

- Definition
- Terminology
- Uses
- Context
Definition

A procedure to find pairs of records in two files that represent the same entity.
Definition

- A procedure to find pairs of records in two files that represent the same entity.
- When both files are the same file, the procedure is to find duplicate records.
Terminology

- Matched records: both records represent the same entity in truth
Terminology

- Matched records: both records represent the same entity in truth
- Linked records: Both records are identified by record linkage procedure as probably representing the same entity
Uses

- Updating and deduplicating a survey frame
Uses

- Updating and deduplicating a survey frame
- Merging two files for microdata analysis
Uses

- Updating and deduplicating a survey frame
- Merging two files for microdata analysis
- Determine confidentiality of microdata
Uses

- Updating and deduplicating a survey frame
- Merging two files for microdata analysis
- Determine confidentiality of microdata
- Measure a population by capture-recapture
Capture-Recapture

Let $A, B$ be independent random samples of sample space $S$

\[
x_{11} = |A \cap B| \quad x_{10} = |A \setminus B| \\
x_{01} = |B \setminus A| \quad x_{00} = |S \setminus (A \cup B)|
\]
Let $A, B$ be independent random samples of sample space $S$

$$x_{11} = |A \cap B| \quad x_{10} = |A \setminus B|$$
$$x_{01} = |B \setminus A| \quad x_{00} = |S \setminus (A \cup B)|$$

Then

$$\hat{x}_{00} = E[x_{00}] = \frac{x_{11} + x_{10} + 1}{x_{11}}$$
Capture-Recapture, Cont.

Take two independent surveys of a region and estimate the number of people missed.
Capture-Recapture, Cont.

- Take two independent surveys of a region and estimate the number of people missed.
- Note accuracy of estimate depends on accuracy of $x_{11}$, as determined by record linkage.
Capture-Recapture, Cont.

- Take two independent surveys of a region and estimate the number of people missed.

- Note accuracy of estimate depends on accuracy of $x_{11}$, as determined by record linkage.

Record Linkage Basics

- Context
- Deterministic Record Linkage
Record Linkage Basics

- Context
- Deterministic Record Linkage
- Probabilistic Record Linkage
Record Linkage Basics

- Context
- Deterministic Record Linkage
- Probabilistic Record Linkage
- Not Statistical Matching
Record Linkage Basics

- Context
- Deterministic Record Linkage
- Probabilistic Record Linkage
- Not Statistical Matching
- Need for Automated Record Linkage
Files have records of fixed length with fields of fixed length and position (or in a database with retrievable individual fields)
Context

- Files have records of fixed length with fields of fixed length and position (or in a database with retrievable individual fields)
- Not a search algorithm
Deterministic Record Linkage

Records are linked when
Deterministic Record Linkage

- Records are linked when
  - They agree exactly on all matching fields
Deterministic Record Linkage

- Records are linked when
  - They agree exactly on all matching fields
  - Or on predetermined portion of fields
Probabilistic Record Linkage

Assign a probabilistic weighting to record pairs
Probabilistic Record Linkage

- Assign a probabilistic weighting to record pairs
- Accepts record pairs with sufficiently high weights as linked pairs
Not Statistical Matching

Statistical matching: Bring together pairs of records with statistically similar characteristics, not necessarily representing the same entity.
Not Statistical Matching

- Statistical matching: Bring together pairs of records with statistically similar characteristics, not necessarily representing the same entity.
- Usually for two files that represent different samples of a population.
Not Statistical Matching

- Statistical matching: Bring together pairs of records with statistically similar characteristics, not necessarily representing the same entity.
- Usually for two files that represent different samples of a population.
- Older practice than exact matching (deterministic or probabilistic).
Need for Automated Record Linkage

Clerical matching is:
Need for Automated Record Linkage

Clerical matching is:

- expensive
Need for Automated Record Linkage

Clerical matching is:

- expensive
- slow
Need for Automated Record Linkage

Clerical matching is:

- expensive
- slow
- error prone
### Need for Automated Record Linkage

Clerical matching is:
- **expensive**
- **slow**
- **error prone**

<table>
<thead>
<tr>
<th></th>
<th>Clerical</th>
<th>1988</th>
<th>1990</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer match proportion</td>
<td>0%</td>
<td>70%</td>
<td>75%</td>
</tr>
<tr>
<td># clerks</td>
<td>3000</td>
<td>600</td>
<td>200</td>
</tr>
<tr>
<td>#months</td>
<td>6</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>False match rate</td>
<td>5%</td>
<td>0.5%</td>
<td>0.2%</td>
</tr>
</tbody>
</table>
Basic Definitions and Notation
Basic Definitions and Notation
Agreement Patterns
Basic Definitions and Notation
Agreement Patterns
Example Comparison Space
Basic Definitions and Notation
Agreement Patterns
Example Comparison Space
Conditional Probabilities
Rec. Link. Theory: Fellegi & Sunter

- Basic Definitions and Notation
- Agreement Patterns
- Example Comparison Space
- Conditional Probabilities
- Linkage Rule
Basic Definitions and Notation
Agreement Patterns
Example Comparison Space
Conditional Probabilities
Linkage Rule
Error Rates
Rec. Link. Theory: Fellegi & Sunter

Basic Definitions and Notation
Agreement Patterns
Example Comparison Space
Conditional Probabilities
Linkage Rule
Error Rates
Clerical Region
Basic Definitions and Notation
Agreement Patterns
Example Comparison Space
Conditional Probabilities
Linkage Rule
Error Rates
Clerical Region
Fundamental Theorem
Basic Definitions and Notation
Agreement Patterns
Example Comparison Space
Conditional Probabilities
Linkage Rule
Error Rates
Clerical Region
Fundamental Theorem
Conditional Independence Assumption
Basic Definitions and Notation

- Sets of entities $A, B$
Basic Definitions and Notation

- Sets of entities \( A, B \)
- Corresponding files of records \( \alpha(A), \beta(B) \)
Basic Definitions and Notation

- Sets of entities $A, B$
- Corresponding files of records $\alpha(A), \beta(B)$
- Sample space $\alpha(A) \times \beta(B)$
Basic Definitions and Notation

- Sets of entities $A, B$
- Corresponding files of records $\alpha (A), \beta (B)$
- Sample space $\alpha (A) \times \beta (B)$
- Matches

$$M = \{(\alpha (a), \beta (b)) \mid a = b\}$$
Basic Definitions and Notation

- Sets of entities $A, B$
- Corresponding files of records $\alpha(A), \beta(B)$
- Sample space $\alpha(A) \times \beta(B)$
- Matches
  \[ M = \{(\alpha(a), \beta(b)) \mid a = b\} \]
- Nonmatches
  \[ U = \{(\alpha(a), \beta(b)) \mid a \neq b\} \]
Basic Definitions and Notation

- Sets of entities $A, B$
Basic Definitions and Notation

- Sets of entities $A, B$
- Corresponding files of records $\alpha(A), \beta(B)$
Basic Definitions and Notation

- Sets of entities $A, B$
- Corresponding files of records $\alpha(A), \beta(B)$
- Sample space $\alpha(A) \times \beta(B)$
Basic Definitions and Notation

- Sets of entities $A, B$
- Corresponding files of records $\alpha (A), \beta (B)$
- Sample space $\alpha (A) \times \beta (B)$
- Matches

\[ M = \{ (\alpha (a), \beta (b)) \mid a = b \} \]
Basic Definitions and Notation

- Sets of entities $A, B$
- Corresponding files of records $\alpha(A), \beta(B)$
- Sample space $\alpha(A) \times \beta(B)$
- Matches
  \[ M = \{(\alpha(a), \beta(b)) \mid a = b\} \]
- Nonmatches
  \[ U = \{(\alpha(a), \beta(b)) \mid a \neq b\} \]
Agreement Patterns

Comparison space

\[ \alpha (A) \times \beta (B) \rightarrow \Gamma \]
Agreement Patterns

- Comparison space

\[ \alpha (A) \times \beta (B) \rightarrow \Gamma \]

- Comparison vector

\[ \gamma \in \Gamma \]
Agreement Patterns

- Comparison space
  \[ \alpha (A) \times \beta (B) \rightarrow \Gamma \]

- Comparison vector
  \[ \gamma \in \Gamma \]

- Each component of comparison vector can take on finitely many values, typically two
Agreement Patterns

- Comparison space

\[ \alpha(A) \times \beta(B) \rightarrow \Gamma \]

- Comparison vector

\[ \gamma \in \Gamma \]

- Each component of comparison vector can take on finitely many values, typically two

\[ \gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n) \]

\[ \gamma_i \in \{0, 1\} \]
Example Comparison Space

Consider 3 binary comparisons
Example Comparison Space

Consider 3 binary comparisons

\( \gamma_1 \) pair agrees on last name
Consider 3 binary comparisons

- $\gamma_1$ pair agrees on last name
- $\gamma_2$ pair agrees on first name
Example Comparison Space

Consider 3 binary comparisons

\[ \gamma_1 \] pair agrees on last name

\[ \gamma_2 \] pair agrees on first name

\[ \gamma_3 \] pair agrees on street name
Example Comparison Space

Consider 3 binary comparisons

- $\gamma_1$: pair agrees on last name
- $\gamma_2$: pair agrees on first name
- $\gamma_3$: pair agrees on street name

Sample agreement pattern

$$\gamma = (1, 0, 1)$$
Conditional Probabilities

- Probability that a record pair has agreement pattern $\gamma$, given that it is a match/nonmatch

\[
\begin{align*}
\Pr (\gamma|M) \\
\Pr (\gamma|U)
\end{align*}
\]
Conditional Probabilities

- Probability that a record pair has agreement pattern $\gamma$, given that it is a match/nonmatch
  \[
  \Pr(\gamma|M) \quad \Pr(\gamma|U)
  \]

- Agreement ratio
  \[
  R(\gamma) = \frac{\Pr(\gamma|M)}{\Pr(\gamma|U)}
  \]
Conditional Probabilities

- Probability that a record pair has agreement pattern $\gamma$, given that it is a match/nonmatch
  
  \[
  \Pr(\gamma|M) \quad \Pr(\gamma|U)
  \]

- Agreement ratio
  
  \[
  R(\gamma) = \frac{\Pr(\gamma|M)}{\Pr(\gamma|U)}
  \]

- Conditioned on the unobservable truth
Linkage Rule

Designate a record pair’s status based on its agreement pattern
Linkage Rule

Designate a record pair’s status based on its agreement pattern

- Link
Linkage Rule

Designate a record pair’s status based on its agreement pattern

- Link
- Non-link
Linkage Rule

Designate a record pair’s status based on its agreement pattern

- Link
- Non-link
- Undecided
Linkage Rule

Designate a record pair’s status based on its agreement pattern

- Link
- Non-link
- Undecided

\[ L : \Gamma \to \{ L, N, C \} \]
Error Rates

False match—a linked pair that is not a match
Error Rates

- False match—a linked pair that is not a match
- False nonmatch—a nonlinked pair that is a match
Error Rates

- False match—a linked pair that is not a match
- False nonmatch—a nonlinked pair that is a match
- False match rate—probability that a designated link is a nonmatch
Error Rates

- False match—a linked pair that is not a match
- False nonmatch—a nonlinked pair that is a match
- False match rate—probability that a designated link is a nonmatch

$$\mu = \Pr (L|U)$$
Error Rates

- False match—a linked pair that is not a match
- False nonmatch—a nonlinked pair that is a match
- False match rate—probability that a designated link is a nonmatch
  \[ \mu = \Pr (L|U) \]
- False nonmatch rate—probability that a designated nonlink is a match
Error Rates

- False match—a linked pair that is not a match
- False nonmatch—a nonlinked pair that is a match
- False match rate—probability that a designated link is a nonmatch
  \[ \mu = \Pr (L|U) \]
- False nonmatch rate—probability that a designated nonlink is a match
  \[ \lambda = \Pr (N|M) \]
Error Rates, Cont.

If all pairs of records are designated link or nonlink
Error Rates, Cont.

If all pairs of records are designated link or nonlink

<table>
<thead>
<tr>
<th></th>
<th>Match</th>
<th>Nonmatch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>$1 - \lambda$</td>
<td>$\mu = \Pr(L</td>
</tr>
<tr>
<td>Nonlink</td>
<td>$\lambda = \Pr(N</td>
<td>M)$</td>
</tr>
</tbody>
</table>
The set $C$ of record pairs which are designated neither probable link nor probable nonlink by the linkage rule.
Clerical Region

- The set $C$ of record pairs which are designated neither probable link nor probable nonlink by the linkage rule.
- The match status of these pairs is left to clerical review.
Fundamental Theorem

Fundamental Theorem


- Order the comparison vectors \( \{ \gamma^j \} \) by their agreement ratios \( R(\gamma^j) \)
Fundamental Theorem


- Order the comparison vectors \( \{ \gamma^j \} \) by their agreement ratios \( R(\gamma^j) \)

- Choose upper \( T_\mu \) and lower \( T_\lambda \) cutoff values for \( R(\gamma) \)
Linkage rule:
Linkage rule:

- Pairs with \( R(\gamma^j) \geq T_\mu \) are designated links
Fundamental Theorem, Cont.

Linkage rule:

- Pairs with $R(\gamma^j) \geq T_\mu$ are designated links
- Pairs with $R(\gamma^j) \leq T_\lambda$ are designated nonlinks
Fundamental Theorem, Cont.

Linkage rule:

- Pairs with $R(\gamma^j) \geq T_\mu$ are designated links
- Pairs with $R(\gamma^j) \leq T_\lambda$ are designated nonlinks
- Pairs with $T_\lambda < R(\gamma^j) < T_\mu$ are in the clerical region
Fundamental Theorem, Cont.

The error rates for this linkage rule are
The error rates for this linkage rule are

\[ \mu = \sum \Pr (\gamma^j | U) \]

subject to \( R(\gamma^j) \geq T_\mu \).
The error rates for this linkage rule are

\[ \mu = \sum_{R(\gamma^j) \geq T_\mu} \Pr (\gamma^j | U) \]

\[ \lambda = \sum_{R(\gamma^j) \leq T_\lambda} \Pr (\gamma^j | M) \]
Fundamental Theorem, Cont.

Theorem: For these error rates $\mu, \lambda$, this is the optimal linkage rule, in the sense of producing the minimum size critical region.
Theorem: For these error rates $\mu$, $\lambda$, this is the optimal linkage rule, in the sense of producing the minimum size critical region.

In other words, for a given error bound tolerance, this rule makes as many linkage decisions as possible.
Weight Distribution for Matches
Weight Distribution for Matches

$$\Pr (w(\gamma) > w_0 | M)$$
Weight Distribution for Non-Matches

weight

USCENSUSBUREAU
Weight Distribution for Non-Matches

\[ \Pr (w(\gamma) < w_0 | U) \]
Idealized Distributions
Idealized Distributions

$T_\lambda$  $T_\mu$  weight
Idealized Distributions

Non-Links | Clerical | Links

$T_\lambda$ | $T_\mu$

weight
Error Rates, Clerical Review Region
Error Rates, Clerical Review Region

![Error Rates Diagram](image-url)
Error Rates, Clerical Review Region

\[ \lambda = \Pr (w (\gamma) < T_{\lambda} | M) \]

\[ \mu = \Pr (w (\gamma) > T_{\mu} | U) \]

\[ T_{\lambda} \]

\[ T_{\mu} \]
To facilitate computation of conditional probabilities, Fellegi & Sunter assume conditional independence of comparison vector components.
Conditional Independence Assumption

To facilitate computation of conditional probabilities, Fellegi & Sunter assume conditional independence of comparison vector components.

For $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$, assume

$$
\Pr(\gamma|M) = \prod_{i=1}^{n} \Pr(\gamma_i|M)
$$

$$
\Pr(\gamma|U) = \prod_{i=1}^{n} \Pr(\gamma_i|U)
$$
The factors \( \Pr (\gamma_i | M) \), \( \Pr (\gamma_i | U) \) are called marginal probabilities.
Cond. Indep. Assumption, Cont.

- The factors $\Pr(\gamma_i | M)$, $\Pr(\gamma_i | U)$ are called **marginal probabilities**

- The ratio

$$
\frac{\Pr(\gamma_i | M)}{\Pr(\gamma_i | U)}
$$

determines the **distinguishing power** of the comparison $\gamma_i$
Under conditional independence assumption, it is convenient to compute the *weight* of the comparison vector.
Under conditional independence assumption, it is convenient to compute the weight of the comparison vector:

\[
\begin{align*}
    w(\gamma) &= \log R(\gamma) \\
    &= \sum_{i=1}^{n} \frac{\log \Pr(\gamma_i|M)}{\log \Pr(\gamma_i|U)} \\
    &= \sum_{i=1}^{n} w(\gamma_i)
\end{align*}
\]
Motivation: Reduce the number of parameters
Motivation: Reduce the number of parameters

For \( n \) binary comparisons and two conditional classes \( M, U \), there are \( 2^{n+1} \) parameters
Motivation: Reduce the number of parameters

For $n$ binary comparisons and two conditional classes $M, U$, there are $2^{n+1}$ parameters

$2^n$ comparison vectors
Motivation: Reduce the number of parameters

For $n$ binary comparisons and two conditional classes $M, U$, there are $2^{n+1}$ parameters

- $2^n$ comparison vectors
- 2 conditional probabilities for each vector
Motivation: Reduce the number of parameters

For \( n \) binary comparisons and two conditional classes \( M, U \), there are \( 2^{n+1} \) parameters

- \( 2^n \) comparison vectors
- 2 conditional probabilities for each vector

Under conditional independence assumption, there are \( 2^n \) parameters
Cond. Indep. Assumption, Cont.

- **Motivation:** Reduce the number of parameters

- For $n$ binary comparisons and two conditional classes $M, U$, there are $2^{n+1}$ parameters
  - $2^n$ comparison vectors
  - 2 conditional probabilities for each vector

- Under conditional independence assumption, there are $2n$ parameters

- **Rationale:** Given $M$, errors producing disagreement should be random
Cond. Indep. Assumption, Cont.

- Often computable in closed form
Cond. Indep. Assumption, Cont.

- Often computable in closed form
- Can produce good decision rules even if model inaccurate
Cond. Indep. Assumption, Cont.

- Often computable in closed form
- Can produce good decision rules even if model inaccurate
- Referred to as naive Bayes in machine learning
Conditional Independence Example

Suppose

\[ \Pr (\gamma_1 = 1 | M) = 0.9 \]
\[ \Pr (\gamma_2 = 1 | M) = 0.8 \]
\[ \Pr (\gamma_3 = 1 | M) = 0.7 \]
Conditional Independence Example

Suppose

\[
\begin{align*}
\Pr(\gamma_1 = 1|M) &= 0.9 \\
\Pr(\gamma_2 = 1|M) &= 0.8 \\
\Pr(\gamma_3 = 1|M) &= 0.7
\end{align*}
\]

Then for \( \gamma = (1, 0, 1) \),

\[
\Pr(\gamma|M) = 0.9 \times 0.2 \times 0.7 = 0.126
\]
Suppose

\[ \Pr (\gamma_1 | M) = 0.8 \quad \Pr (\gamma_1 | U) = 0.1 \]
\[ \Pr (\gamma_2 | M) = 0.9 \quad \Pr (\gamma_2 | U) = 0.3 \]
Conditional Independence Example

Suppose

\[ \Pr (\gamma_1|M) = 0.8 \quad \Pr (\gamma_1|U) = 0.1 \]
\[ \Pr (\gamma_2|M) = 0.9 \quad \Pr (\gamma_2|U) = 0.3 \]

Then

\[ \frac{\Pr (\gamma_1|M)}{\Pr (\gamma_1|U)} = 8.0 \]
\[ \frac{\Pr (\gamma_2|M)}{\Pr (\gamma_2|U)} = 3.0 \]
Fellegi-Sunter Summary

Choose conditional probability parameters
Fellegi-Sunter Summary

- Choose conditional probability parameters
- Conduct field comparisons on record pairs
Fellegi-Sunter Summary

- Choose conditional probability parameters
- Conduct field comparisons on record pairs
- Classify record pairs based on weight of comparison vector
Record Linkage Methodology

Parameter estimation
Record Linkage Methodology

- Parameter estimation
- EM Algorithm
Record Linkage Methodology

- Parameter estimation
  - EM Algorithm
- Blocking
Choosing Parameters

Informal
Choosing Parameters

- Informal
- EM Algorithm
Choosing Parameters

- Informal
- EM Algorithm
- Other Methods
Informal Methods

- Guess
Informal Methods

Answer: Guess

\[ 0 < \Pr(\gamma|U) < \Pr(\gamma|M) < 1 \]
Informal Methods

- Guess

\[ 0 < \Pr (\gamma|U) < \Pr (\gamma|M) < 1 \]

- Approximate
Informal Methods

- **Guess**

\[ 0 < \Pr(\gamma|U) < \Pr(\gamma|M) < 1 \]

- **Approximate**

\[ \Pr(\gamma|U) \approx \Pr(\gamma|S) \]
Informal Methods

- Guess
  \[ 0 < \Pr (\gamma | U) < \Pr (\gamma | M) < 1 \]

- Approximate
  \[ \Pr (\gamma | U) \approx \Pr (\gamma | S) \]

- Iterate
Informal Methods

- **Guess**
  
  \[0 < \Pr(\gamma|U) < \Pr(\gamma|M) < 1\]

- **Approximate**
  
  \[\Pr(\gamma|U) \approx \Pr(\gamma|S)\]

- **Iterate**
  
  Perform matching with current parameters
Informal Methods

- **Guess**

\[ 0 < \Pr(\gamma|U) < \Pr(\gamma|M) < 1 \]

- **Approximate**

\[ \Pr(\gamma|U) \approx \Pr(\gamma|S) \]

- **Iterate**
  - Perform matching with current parameters
  - Review results
Informal Methods

- **Guess**

  \[0 < \Pr(\gamma|U) < \Pr(\gamma|M) < 1\]

- **Approximate**

  \[\Pr(\gamma|U) \approx \Pr(\gamma|S)\]

- **Iterate**
  - Perform matching with current parameters
  - Review results
  - Adjust parameters based on observation
EM Algorithm

EM Algorithm


EM Algorithm


- Maximum likelihood method
EM Algorithm


- Maximum likelihood method

- Latent class
EM Algorithm


- Maximum likelihood method
- Latent class
- Mixture model
Likelihood Function

\[
L = \prod_{(a,b) \in S} \Pr (\gamma (a, b))
\]

\[
= \prod_j \left( \Pr (\gamma^j | M) \Pr (M) + \Pr (\gamma^j | U) \Pr (U) \right)^{n_j}
\]
Likelihood Function

\[ L = \prod_{(a,b) \in S} \Pr(\gamma(a, b)) \]

\[ = \prod_j \left( \Pr(\gamma^j | M) \Pr(M) + \Pr(\gamma^j | U) \Pr(U) \right)^{n_j} \]

\[ n_j = \left| \left\{ (a, b) \in S \mid \gamma(a, b) = \gamma^j \right\} \right| \]
Consider

\[
\chi_j = \begin{cases} 
1 & \text{if } (a, b)^j \in M \\
0 & \text{if } (a, b)^j \in U
\end{cases}
\]

\[
X_j = \sum_{\gamma(a, b) = \gamma^j} \chi_j (a, b)
\]
Consider

\[ \chi_j = \begin{cases} 
1 & \text{if } (a, b)^j \in M \\
0 & \text{if } (a, b)^j \in U 
\end{cases} \]

\[ X_j = \sum_{\gamma(a,b)=\gamma^j} \chi_j (a, b) \]

Then

\[ L = \prod_j \left( (\Pr (\gamma^j | M) \Pr (M))^x_j \left( (\Pr (\gamma^j | U) \Pr (U))^{1-x_j} \right)^{n_j} \right) \]
Expectation Step

- Given current estimates of conditional probabilities and $Pr(M), Pr(U)$, compute
Expectation Step

Given current estimates of conditional probabilities and $\Pr(M)$, $\Pr(U)$, compute

$$E\left(\frac{X^j}{\gamma^j}\right) = \Pr(M|\gamma^j)$$

$$\quad = \frac{\Pr(\gamma^j|M) \Pr(M)}{\Pr(\gamma^j|M) \Pr(M) + \Pr(\gamma^j|U) \Pr(U)}$$

$$\quad = \hat{X}^j$$
Maximization Step

Given unobserved data estimates $\hat{X}^j$, compute probabilities $\Pr (\gamma^j | M)$, $\Pr (\gamma^j | U)$, $\Pr (M)$, $\Pr (U)$ maximizing
Maximization Step

Given unobserved data estimates $\hat{X}^j$, compute probabilities $\Pr(\gamma^j|M)$, $\Pr(\gamma^j|U)$, $\Pr(M)$, $\Pr(U)$ maximizing

$$\log L = \sum_j n_j \left( \hat{X}^j \left( \log \Pr(\gamma^j|M) + \log \Pr(M) \right) ight)$$

$$+ \left( 1 - \hat{X}^j \right) \left( \log \Pr(\gamma^j|U) + \log \Pr(U) \right)$$
Max Step, Cont.

Under conditional independence
Max Step, Cont.

Under conditional independence

\[ \log L = \sum_j n_j \left( \sum_i \hat{X}^j \left( \log \Pr \left( \gamma_i^j | M \right) + \log \Pr \left( M \right) \right) \right) 
+ \left( 1 - \hat{X}^j \right) \left( \sum_i \log \Pr \left( \gamma_i^j | U \right) + \log \Pr \left( U \right) \right) \]
Max Step, Cont.

For

\[ n = \sum_j n_j \]

estimate

\[ \Pr (M) = \frac{1}{n} \sum_j n_j \bar{X}^j \]
Max Step, Cont.

Let

\[ k_{i}^{j} = \begin{cases} 1 & \text{if } \gamma_{i}^{j} = 1 \\ 0 & \text{if } \gamma_{i}^{j} = 0 \end{cases} \]
Let

\[ k^j_i = \begin{cases} 
1 & \text{if } \gamma^j_i = 1 \\
0 & \text{if } \gamma^j_i = 0
\end{cases} \]

and estimate

\[ \Pr (\gamma_i | M) = \frac{1}{n} \sum_j n_j \bar{X}_j^j k^j_i \]
EM Algorithm

1. Initialize with probability values
EM Algorithm

1. Initialize with probability values
2. Iterate
EM Algorithm

1. Initialize with probability values
2. Iterate
   (a) Expectation Step
EM Algorithm

1. Initialize with probability values
2. Iterate
   (a) Expectation Step
   (b) Maximization Step
EM Algorithm

1. Initialize with probability values
2. Iterate
   (a) Expectation Step
   (b) Maximization Step
3. Until convergence of likelihood function
EM Algorithm Remarks

- Each EM iteration increases likelihood, so algorithm converges to a (local) maximum
EM Algorithm Remarks

- Each EM iteration increases likelihood, so algorithm converges to a (local) maximum.
- For this conditional independence model, convergence is efficient and generally insensitive to initial data.
EM Algorithm Remarks

- Each EM iteration increases likelihood, so algorithm converges to a (local) maximum.
- For this conditional independence model, convergence is efficient and generally insensitive to initial data.
- For latent class to be numerically detected, it must be represented by about 5% of the total record pair data.
EM Algorithm Remarks

- Each EM iteration increases likelihood, so the algorithm converges to a (local) maximum.
- For this conditional independence model, convergence is efficient and generally insensitive to initial data.
- For latent class to be numerically detected, it must be represented by about 5% of the total record pair data.
- Check: Do $\Pr(M), \Pr(U)$ seem reasonable?
If $Pr(M), Pr(U)$ are off, everything is off.
EM Remarks, Cont.

- If $\Pr(M), \Pr(U)$ are off, everything is off.
- We can extend problem to comparisons taking on more than 2 values.
EM Remarks, Cont.

- If $\Pr(M), \Pr(U)$ are off, everything is off
- We can extend problem to comparisons taking on more than 2 values
  - Creates more pattern types and probability parameters
EM Remarks, Cont.

- If $\Pr(M)$, $\Pr(U)$ are off, everything is off
- We can extend problem to comparisons taking on more than 2 values
  - Creates more pattern types and probability parameters
- Can extend algorithm to more than 2 classes
EM Remarks, Cont.

- If $\Pr(M)$, $\Pr(U)$ are off, everything is off.
- We can extend problem to comparisons taking on more than 2 values.
  - Creates more pattern types and probability parameters.
- Can extend algorithm to more than 2 classes.
  - Increases number of parameters to be estimated.
Blocking

If set $A$ contains $m$ records and set $B$ contains $n$ records then $A \times B$ contains $mn$ record pairs.
Blocking

- If set $A$ contains $m$ records and set $B$ contains $n$ records then $A \times B$ contains $mn$ record pairs.

- It is computationally inefficient to compare all record pairs.
Blocking

- If set $A$ contains $m$ records and set $B$ contains $n$ records then $A \times B$ contains $mn$ record pairs.
- It is computationally inefficient to compare all record pairs.
- In practice, just bring together record pairs that agree on some chosen features (blocking criterion).
Blocking

- If set $A$ contains $m$ records and set $B$ contains $n$ records then $A \times B$ contains $mn$ record pairs.

- It is computationally inefficient to compare all record pairs.

- In practice, just bring together record pairs that agree on some chosen features (blocking criterion).

- Generally repeat record linkage procedure for several different blocking criteria.
Blocking Criteria

- Geographic codes
Blocking Criteria

- Geographic codes
- Postal or phone codes
Blocking Criteria

- Geographic codes
- Postal or phone codes
- Name prefix
Blocking Criteria

- Geographic codes
- Postal or phone codes
- Name prefix
- Phonetic name codes
Blocking Criteria

- Geographic codes
- Postal or phone codes
- Name prefix
- Phonetic name codes
  - Soundex
Blocking Criteria

- Geographic codes
- Postal or phone codes
- Name prefix
- Phonetic name codes
  - Soundex
  - NYSIIS
Blocking Criteria

- Geographic codes
- Postal or phone codes
- Name prefix
- Phonetic name codes
  - Soundex
  - NYSIIS
- Combinations
Record Linkage Refinements

- String comparator
Record Linkage Refinements

- String comparator
- Third latent class
Record Linkage Refinements

- String comparator
- Third latent class
- Third comparison value
Record Linkage Refinements

- String comparator
- Third latent class
- Third comparison value
- One-to-one matching
String Comparator

For some comparisons (e.g. categorical variables), it is sufficient to assign agree/disagree
String Comparator

- For some comparisons (e.g. categorical variables), it is sufficient to assign agree/disagree.
- For string variables (e.g. first names, last names, street names) this is probably too restrictive.
String Comparator

- For some comparisons (e.g. categorical variables), it is sufficient to assign agree/disagree.

- For string variables (e.g. first names, last names, street names) this is probably too restrictive.

- A string comparator allows us to assign comparison values between full agreement and full disagreement.
Binary comparison $\gamma \in \{0, 1\}$
String Comparator Context

- Binary comparison $\gamma \in \{0, 1\}$
- Weight assignment

$$a_w = \log \frac{\Pr (\gamma = 1|M)}{\Pr (\gamma = 1|U)}$$

$$d_w = \log \frac{\Pr (\gamma = 0|M)}{\Pr (\gamma = 0|U)}$$

$d_w < 0 < a_w$
For alphabet $\Sigma$, our string comparator is a similarity function

$$\gamma : \Sigma^* \times \Sigma^* \rightarrow [0, 1]$$

$$\gamma (\alpha, \beta) = 1 \text{ if } \alpha = \beta$$
For alphabet $\Sigma$, our string comparator is a similarity function

$$\gamma : \Sigma^* \times \Sigma^* \to [0, 1]$$

$$\gamma (\alpha, \beta) = 1 \text{ if } \alpha = \beta$$

Weight assignment function $w$ is an increasing interpolation function

$$w : [0, 1] \to [d_w, a_w]$$

$$w (1) = a_w$$
Some String Comparator Types

- Bigram, \( n \)-gram
Some String Comparator Types

- Bigram, $n$-gram
- Jaro-Winkler
Some String Comparator Types

- Bigram, $n$-gram
- Jaro-Winkler
- Edit distance
Bigrams

- Decompose string into a set of 2-character (contiguous) substrings

\[ \text{alphabet} \rightarrow \{al, lp, ph, ha, ab, be, et\} \]
Bigrams

- Decompose string into a set of 2-character (contiguous) substrings

\[ \text{alphabet} \rightarrow \{al, lp, ph, ha, ab, be, et\} \]

- For alphabet of \( s = |\Sigma| \) characters, record bigram counts in a vector of dimension \( s^2 \)
Two strings can be compared by computing the “angle” between their bigram vectors $a, b$

$$\cos \theta = \frac{a \cdot b}{|a| |b|}$$
Bigrams, Cont.

Two strings can be compared by computing the “angle” between their bigram vectors $a, b$

$$\cos \theta = \frac{a \cdot b}{|a||b|}$$

Obvious generalization to $n$-grams
Bigrams, Cont.

- Two strings can be compared by computing the “angle” between their bigram vectors $a, b$

$$\cos \theta = \frac{a \cdot b}{|a| |b|}$$

- Obvious generalization to $n$-grams
- Vector for $n$-gram is in $s^n$ dimensional space
Computation algorithm is fast (linear)
Bigrams, Cont.

- Computation algorithm is fast (linear)
- Don’t work very well for record linkage
Bigrams, Cont.

- Computation algorithm is fast (linear)
- Don’t work very well for record linkage
  - Ignores order of bigram occurrence

\[ abcba \approx bcbab \]
Bigrams, Cont.

- Computation algorithm is fast (linear)
- Don’t work very well for record linkage
  - Ignores order of bigram occurrence
    $abcba \approx bcbab$
  - Works better for small alphabet, long strings than *vice versa*
Jaro-Winkler Comparator

In the following, let

\[ \alpha = (a_1, a_2, \ldots, a_m), \beta = (b_1, b_2, \ldots, b_n) \]

be strings of lengths \( m, n \) respectively with \( m \leq n \)
In the following, let
\[ \alpha = (a_1, a_2, \ldots a_m), \beta = (b_1, b_2, \ldots, b_n) \]
be strings of lengths \( m, n \) respectively with \( m \leq n \).

Comparator value depends on number of common characters and character “transpositions”
Strings $\alpha, \beta$ have common characters $a_i, b_j$ iff

$$a_i = b_j$$

$$|i - j| < \left\lfloor \frac{n}{2} \right\rfloor$$
Strings $\alpha, \beta$ have common characters $a_i, b_j$ iff

\[ a_i = b_j \]

\[ |i - j| < \left\lfloor \frac{n}{2} \right\rfloor \]

The number of transpositions is computed as the greatest integer of half of the number of out-of-order common character pairs.
For string pair with $c$ common characters and $t$ transpositions, basis similarity score is

$$x = \frac{1}{3} \left( \frac{c}{m} + \frac{c}{n} + \frac{c - t}{c} \right)$$
Consider the strings \( (b,a,r,n,e,s) \) and \( (a,n,d,e,r,s,o,n) \)
Jaro-Winkler Example

- Consider the strings \((b,a,r,n,e,s)\) and \((a,n,d,e,r,s,o,n)\)
- Search range \(d\)

\[
n = 8 \\
d = \left\lfloor \frac{8}{2} \right\rfloor - 1 = 3
\]
Jaro-Winkler Example

Consider the strings \((b,a,r,n,e,s)\) and \((a,n,d,e,r,s,o,n)\)

Search range \(d\)

\[
\begin{align*}
n &= 8 \\
d &= \left\lfloor \frac{8}{2} \right\rfloor - 1 = 3
\end{align*}
\]

Common characters

\((a, r, n, e, s)\)

\((a, n, e, r, s)\)
Jaro-Winkler Example, Cont.

Five common characters with 3 out of order, so \( c = 5, t = 1 \)
Jaro-Winkler Example, Cont.

- Five common characters with 3 out of order, so \( c = 5, t = 1 \)

- Score

\[
x = \frac{1}{3} \left( \frac{5}{6} + \frac{5}{8} + \frac{4}{5} \right) = \frac{271}{360} = 0.75280
\]
Jaro-Winkler Variations

- Similar characters
Jaro-Winkler Variations

- Similar characters
- Prefix adjustment
Jaro-Winkler Variations

- Similar characters
- Prefix adjustment
- Long suffix adjustment
Similar Characters

- Attempt to compensate for common misspellings or typos
Similar Characters

- Attempt to compensate for common misspellings or typos
- List of 36 pairs of characters deemed similar (e.g. most vowel pairs)
Similar Characters

- Attempt to compensate for common misspellings or typos
- List of 36 pairs of characters deemed similar (e.g. most vowel pairs)
- After common characters designated, remaining characters checked for similar pairs
Similar Characters

- Attempt to compensate for common misspellings or typos
- List of 36 pairs of characters deemed similar (e.g. most vowel pairs)
- After common characters designated, remaining characters checked for similar pairs
- Each similar pair is scored as 0.3 of a common pair
Revised character count

\[ c_s = c + 0.3s \]
Similar Characters, Cont.

- Revised character count
  
  \[ c_s = c + 0.3s \]

- Adjusted comparator score
  
  \[ x_s = \frac{1}{3} \left( \frac{c_s}{m} + \frac{c_s}{n} + \frac{c - t}{c} \right) \]
Similar Characters, Cont.

For example. $abc$ and $ebc$ have 2 common characters and the remaining pair $(a,e)$ are similar, so

$$x_s = \frac{1}{3} \left( \frac{2}{3} + \frac{2}{3} + 1 \right) + \frac{1}{3} \left( \frac{0.3}{3} + \frac{0.3}{3} \right)$$

$$= \frac{7}{9} + \frac{1}{15}$$

$$= \frac{38}{45}$$
Common Prefix

- Spelling mistakes tend to occur later in the string (Winkler)
Common Prefix

- Spelling mistakes tend to occur later in the string (Winkler)
- Check for common prefix of up to 4 characters
Common Prefix

- Spelling mistakes tend to occur later in the string (Winkler)
- Check for common prefix of up to 4 characters
- If length of common prefix is $p$, adjust score $x$ by

$$x_p = x + \frac{p(1 - x)}{10}$$
Long String Adjustment

- Adjust score for longer strings with several common characters beyond common prefix
Long String Adjustment

- Adjust score for longer strings with several common characters beyond common prefix
- Conditions for using the adjustment
Long String Adjustment

- Adjust score for longer strings with several common characters beyond common prefix

- Conditions for using the adjustment
  1. \( m \geq 5 \)
Long String Adjustment

- Adjust score for longer strings with several common characters beyond common prefix

- Conditions for using the adjustment
  1. \( m \geq 5 \)
  2. \( c - p \geq 2 \)
Long String Adjustment

- Adjust score for longer strings with several common characters beyond common prefix

- Conditions for using the adjustment
  1. $m \geq 5$
  2. $c - p \geq 2$
  3. $c - p \geq \frac{m - p}{2}$
Long String Adjustment, Cont.

That is,
That is,

1. Both strings are at least 5 characters long
That is,

1. Both strings are at least 5 characters long
2. There are at least two common characters besides the agreeing prefix characters
That is,

1. Both strings are at least 5 characters long
2. There are at least two common characters besides the agreeing prefix characters
3. We want the strings outside the common prefixes to be fairly rich in common characters, so that the remaining common characters are at least half of the remaining common characters of the shorter string
If conditions met, then adjust score by

\[ x_l = x + (1 - x) \frac{c - (p + 1)}{m + n - 2(p - 1)} \]
In *barnes, anderson* example, conditions are met, so the adjusted score is

\[ x_l = \frac{271}{360} + \left(1 - \frac{271}{360}\right) \frac{5 - 1}{6 + 8 + 2} \]

\[ = \frac{391}{480} \]

\[ \approx 0.8146 \]
Jaro-Winkler Comparator

- Slower algorithm (quadratic)
Jaro-Winkler Comparator

- Slower algorithm (quadratic)
- Performs very well in tests
Edit Distance String Comparators

The minimum number of edits required to convert string $\alpha$ to string $\beta$, lengths $m \leq n$
The minimum number of edits required to convert string $\alpha$ to string $\beta$, lengths $m \leq n$

- Insert
The minimum number of edits required to convert string $\alpha$ to string $\beta$, lengths $m \leq n$

- Insert
- Delete
The minimum number of edits required to convert string $\alpha$ to string $\beta$, lengths $m \leq n$:
- Insert
- Delete
- Substitute
Edit Distance String Comparators

- The minimum number of edits required to convert string $\alpha$ to string $\beta$, lengths $m \leq n$
- Insert
- Delete
- Substitute
- Dynamic programming algorithm, quadratic complexity $O(mn)$
For $\alpha_i$ prefix of $\alpha$ of length $i$, $\beta_j$ prefix of $\beta$ of length $j$
Edit Distance Algorithm

For $\alpha_i$ prefix of $\alpha$ of length $i$, $\beta_j$ prefix of $\beta$ of length $j$

Initialize

\[
e(\alpha_i, \varepsilon) = i
\]
\[
e(\varepsilon, \beta_j) = j
\]
\[
e(\varepsilon, \varepsilon) = 0
\]
Edit Distance Algorithm, Cont.

Compute

\[
e(\alpha_i, \beta_j) = \min \left\{ 
  \begin{array}{l}
  e(\alpha_{i-1}, \beta_j) + 1 \\
  e(\alpha_i, \beta_{j-1}) + 1 \\
  e(\alpha_{i-1}, \beta_{j-1}) + 1 \\
  \end{array}
  \right. 
  \begin{array}{l}
  \text{if } a_i = b_j \\
  \text{if } a_i \neq b_j \\
  \end{array}
\]

Record Linkage: Theory and Practice – p. 81/134
Edit Distance Algorithm, Cont.

Compute

\[ e(\alpha_i, \beta_j) = \min \begin{cases} 
  e(\alpha_{i-1}, \beta_j) + 1 \\
  e(\alpha_i, \beta_{j-1}) + 1 \\
  e(\alpha_{i-1}, \beta_{j-1}) & \text{if } a_i = b_j \\
  e(\alpha_{i-1}, \beta_{j-1}) + 1 & \text{if } a_i \neq b_j 
\end{cases} \]

Distance

\[ e = e(\alpha, \beta) = e(\alpha_m, \beta_n) \]
Edit distance is a metric
Edit Distance Similarity Function

- Edit distance is a metric
- Similarity function

\[ x_e = 1 - \frac{e}{n} \]
For example, for *barnes, anderson*, have possible minimal edit path

\[(b, \varepsilon) a (r, n) (n, d) e (\varepsilon, r) s (\varepsilon, o) (\varepsilon, n)\]
Edit Distance Example

For example, for *barnes, anderson*, have possible minimal edit path

\[(b, \varepsilon) \ a \ (r, n) \ (n, d) \ e \ (\varepsilon, r) \ s \ (\varepsilon, o) \ (\varepsilon, n)\]

So

\[x_e = 1 - \frac{6}{8} = \frac{1}{4}\]
Edit Distance Example

For example, for *barnes, anderson*, have possible minimal edit path

\[(b, \varepsilon) a (r, n) (n, d) e (\varepsilon, r) s (\varepsilon, o) (\varepsilon, n)\]

So

\[x_e = 1 - \frac{6}{8} = \frac{1}{4}\]

Note order of characters very important
Longest Common Subsequence

- Length of longest common subsequence ($lcs$)
Longest Common Subsequence

- Length of longest common subsequence (lcs)
- Similar dynamic programming algorithm, without substitutions

\[ e(\alpha_i, \beta_j) = \min \begin{cases} 
  e(\alpha_{i-1}, \beta_j) + 1 \\
  e(\alpha_i, \beta_{j-1}) + 1 \\
  e(\alpha_{i-1}, \beta_{j-1}) & \text{if } a_i = b_j 
\end{cases} \]
LCS Similarity Function

Similarity function

\[ x_c = \frac{l}{m} \]
LCS Similarity Function

- Similarity function

$$x_c = \frac{l}{m}$$

- Example $lcs= (a, n, e, s)$, similarity score

$$x_c = \frac{4}{6} = \frac{2}{3}$$
Combination Similarity Function

- Compute both edit distance and \( lcs \)
Combination Similarity Function

- Compute both edit distance and lcs
- Combined score

\[ x_{ec} = \frac{1}{2} \left( \left( 1 - \frac{e}{n} \right) + \frac{l}{m} \right) \]
Combination Similarity Function

- Compute both edit distance and LCS
- Combined score

\[ x_{ec} = \frac{1}{2} \left( \left( 1 - \frac{e}{n} \right) + \frac{l}{m} \right) \]

- Example

\[ x_{ec} = \frac{1}{2} \left( \frac{1}{4} + \frac{2}{3} \right) = \frac{11}{24} \approx 0.4583 \]
Evaluating String Comparators

Evaluating String Comparators


- Compare performance
Evaluating String Comparators


Compare performance

- Jaro-Winkler
Evaluating String Comparators


- Compare performance
  - Jaro-Winkler
  - Edit distance
Evaluating String Comparators, Cont.

Jaro-Winkler, with and without modifications
Evaluating String Comparators, Cont.

- Jaro-Winkler, with and without modifications
- Prefix adjustment
Evaluating String Comparators, Cont.

- Jaro-Winkler, with and without modifications
  - Prefix adjustment
  - Similar characters
Evaluating String Comparators, Cont.

- Jaro-Winkler, with and without modifications
  - Prefix adjustment
  - Similar characters
  - Long suffix adjustment
Evaluating String Comparators, Cont.

- Edit distance
Evaluating String Comparators, Cont.

- Edit distance
- Edit distance similarity
Evaluating String Comparators, Cont.

- Edit distance
- Edit distance similarity
Evaluating String Comparators, Cont.

- Edit distance
- Edit distance similarity
- With and without *lcs*
Lots of data
Evaluating String Comparators, Cont.

- Lots of data
- Truth decks from 1990 and 2000 U.S. Census
Lots of data

Truth decks from 1990 and 2000 U.S. Census

$M$: All non-identical, non-blank names from matched record pairs
Evaluating String Comparators, Cont.

- Lots of data
- Truth decks from 1990 and 2000 U.S. Census
- $M$: All non-identical, non-blank names from matched record pairs
- $U$: All cross pairs of these names
Results of String Comparator Evaluation

- Jaro-Winkler did well
Results of String Comparator Evaluation

- Jaro-Winkler did well
- Prefix adjustment always helps
Results of String Comparator Evaluation

- Jaro-Winkler did well
  - Prefix adjustment always helps
  - Similar character adjustment generally helps a bit
Results of String Comparator Evaluation

- Jaro-Winkler did well
  - Prefix adjustment always helps
  - Similar character adjustment generally helps a bit
  - Long suffix adjustment sometime helps a little
Results of String Comparator Evaluation

- Adding \textit{lcs} significantly improves edit distance and Markov edit distance
Results of String Comparator Evaluation

- Adding \textit{lcs} significantly improves edit distance and Markov edit distance
- Edit distance always better than Markov edit distance
Results of String Comparator Evaluation

- Adding \textit{lcs} significantly improves edit distance and Markov edit distance
- Edit distance always better than Markov edit distance
- Jaro-Winkler (full) comparable to edit distance/\textit{lcs}
Results of String Comparator Evaluation

- Adding \textit{lcs} significantly improves edit distance and Markov edit distance.
- Edit distance always better than Markov edit distance.
- Jaro-Winkler (full) comparable to edit distance/\textit{lcs}.
- Usually
Let $\alpha, \beta$ be strings of length $n$ with no common characters.
Let $\alpha, \beta$ be strings of length $n$ with no common characters

For Jaro-Winkler
Let $\alpha, \beta$ be strings of length $n$ with no common characters

For Jaro-Winkler

\[ s(\alpha, \alpha\beta) = \frac{5}{6} \]
Jaro-Winkler Anomaly

- Let $\alpha, \beta$ be strings of length $n$ with no common characters

- For Jaro-Winkler

  \[
  s(\alpha, \alpha\beta) = \frac{5}{6}
  \]

  In $n \geq 4$, with prefix adjustment,

  \[
  s(\alpha, \alpha\beta) = \frac{9}{10}
  \]
Jaro-Winkler Anomaly

Let $\alpha, \beta$ be strings of length $n$ with no common characters.

For Jaro-Winkler

- $s(\alpha, \alpha\beta) = \frac{5}{6}$
- In $n \geq 4$, with prefix adjustment,
  $s(\alpha, \alpha\beta) = \frac{9}{10}$
- $s(\beta, \alpha\beta) = 0$
Jaro-Winkler Anomaly

Let $\alpha, \beta$ be strings of length $n$ with no common characters.

For Jaro-Winkler

$s(\alpha, \alpha\beta) = \frac{5}{6}$

In $n \geq 4$, with prefix adjustment,

$s(\alpha, \alpha\beta) = \frac{9}{10}$

$s(\beta, \alpha\beta) = 0$

For edit-distance/lcs,

$s(\alpha, \alpha\beta) = s(\beta, \alpha\beta) = \frac{3}{4}$
Hybrid Comparator

Compute both Jaro-Winkler and edit distance/\textit{lcs}
Hybrid Comparator

- Compute both Jaro-Winkler and edit distance/\textit{lcs}
- Use larger of Jaro-Winkler and (scaled) edit distance/\textit{lcs}
Hybrid Comparator

- Compute both Jaro-Winkler and edit distance/lcs
- Use larger of Jaro-Winkler and (scaled) edit distance/lcs
- Where J-W does well, hybrid does a little better than either
Hybrid Comparator

- Compute both Jaro-Winkler and edit distance/\text{lcs}
- Use larger of Jaro-Winkler and (scaled) edit distance/\text{lcs}
- Where J-W does well, hybrid does a little better than either
- Where J-W does significantly worse, hybrid does nearly as well as edit distance/\text{lcs}
Hybrid Comparator, Cont.

Can see some improvement in actual record linkage results
Hybrid Comparator, Cont.

- Can see some improvement in actual record linkage results
- Calculation takes a long time
String Comparator Summary

String comparator improves record linkage
String Comparator Summary

- String comparator improves record linkage
- String comparator takes significant amount of record linkage computation time
String Comparator Summary

- String comparator improves record linkage
- String comparator takes significant amount of record linkage computation time
  - For J-W, about 30%
EM algorithm generalizes to more than 2 classes, $M, U$
EM algorithm generalizes to more than 2 classes, $M, U$

Does $U$ have any natural partitions?
More Than Two Latent Classes

- EM algorithm generalizes to more than 2 classes, $M, U$
- Does $U$ have any natural partitions?
- For Census data
More Than Two Latent Classes

- EM algorithm generalizes to more than 2 classes, $M, U$
- Does $U$ have any natural partitions?
- For Census data
  - $U_1$, different people, same household
More Than Two Latent Classes

EM algorithm generalizes to more than 2 classes, $M, U$

Does $U$ have any natural partitions?

For Census data
- $U_1$, different people, same household
- $U_2$, different people, different household
EM algorithm generalizes to more than 2 classes, $M, U$

Does $U$ have any natural partitions?

For Census data
- $U_1$, different people, same household
- $U_2$, different people, different household

Classes have to be implicit in the matching data
EM for Three Classes

Use EM to estimate $\Pr(U_1)$, $\Pr(U_2)$, and marginal probabilities $\Pr(\gamma_i | U_1)$, $\Pr(\gamma_i | U_2)$
EM for Three Classes

Use EM to estimate $\Pr(U_1), \Pr(U_2)$, and marginal probabilities $\Pr(\gamma_i|U_1), \Pr(\gamma_i|U_2)$

Recombine

$$\Pr(\gamma_i|U) = \frac{\Pr(\gamma_i|U_1) \Pr(U_1) + \Pr(\gamma_i|U_2) \Pr(U_2)}{\Pr(U_1) + \Pr(U_2)}$$
More Than Two Comparison Values

Can have more than \{agree, disagree\}
More Than Two Comparison Values

- Can have more than \{agree, disagree\}
- For $m$ comparison values, EM algorithm must estimate $2(m - 1)$ parameters
More Than Two Comparison Values

- Can have more than \{agree, disagree\}
- For \(m\) comparison values, EM algorithm must estimate \(2(m - 1)\) parameters
- We have used \{agree, disagree, missing\} when data is often missing but has distinguishing power when present
More Than Two Comparison Values

- Can have more than \{agree, disagree\}
- For \(m\) comparison values, EM algorithm must estimate \(2(m - 1)\) parameters
- We have used \{agree, disagree, missing\} when data is often missing but has distinguishing power when present
  - For example, middle initial
Reasonability check for parameter estimation

\[
\log \frac{\Pr \left( \text{blank} \mid M \right)}{\Pr \left( \text{blank} \mid U \right)} \approx 0
\]
One-to-one Matching

If both files have no duplication within them, then it is preferable to have output with each record linked to no more than one record in the other file.
One-to-one Matching

- If both files have no duplication within them, then it is preferable to have output with each record linked to no more than one record in the other file.

- All records that are compared with each other are within a block.
One-to-one Matching

- If both files have no duplication within them, then it is preferable to have output with each record linked to no more than one record in the other file.

- All records that are compared with each other are within a block.

- Linear assignment algorithm used to find optimal one-to-one matches within a block.
Linear Assignment Algorithm

For agreement weights in block

<table>
<thead>
<tr>
<th></th>
<th>$B_1$</th>
<th>$B_2$</th>
<th>$B_3$</th>
<th>\cdots</th>
<th>$B_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$w_{11}$</td>
<td>$w_{12}$</td>
<td>$w_{13}$</td>
<td>$w_{1n}$</td>
<td></td>
</tr>
<tr>
<td>$A_2$</td>
<td>$w_{21}$</td>
<td>$w_{22}$</td>
<td>$w_{23}$</td>
<td>$w_{2n}$</td>
<td></td>
</tr>
<tr>
<td>$A_3$</td>
<td>$w_{31}$</td>
<td>$w_{32}$</td>
<td>$w_{33}$</td>
<td>$w_{3n}$</td>
<td></td>
</tr>
<tr>
<td>\vdots</td>
<td>$w_{n1}$</td>
<td>$w_{n2}$</td>
<td>$w_{n3}$</td>
<td>$w_{nn}$</td>
<td></td>
</tr>
</tbody>
</table>
Linear Assignment Algorithm

Find permutation $\sigma$ that maximizes

$$\sum_{i=1}^{n} w_{i,\sigma(i)}$$
Linear Assignment Algorithm

Find permutation $\bar{\sigma}$ that maximizes

$$\sum_{i=1}^{n} w_{i,\sigma(i)}$$

Not a greedy algorithm
Linear Assignment Algorithm

Find permutation $\sigma$ that maximizes

$$\sum_{i=1}^{n} w_{i,\sigma(i)}$$

Not a greedy algorithm

- Father 40 ↔ Mother 39
- Mother 39 ↔ Daughter 16
- Daughter 16 ↔ Son 13
- Son 13
Error Rates

**False Match Rate**

\[ \mu = \Pr(L | U) = \Pr(w(\gamma) < T_\mu | U) \]
Error Rates

- False Match Rate

\[ \mu = \Pr (L | U) = \Pr (w(\gamma) < T_\mu | U) \]

- False Non-match Rate

\[ \lambda = \Pr (N | M) = \Pr (w(\gamma) > T_\lambda | U) \]
Question: Relative to what sample space?
Practical Considerations

Question: Relative to what sample space?

\[ A \times B \]
Practical Considerations

Question: Relative to what sample space?

$A \times B$

Pairs in blocking scheme
Practical Considerations

Question: Relative to what sample space?
- \( A \times B \)
- Pairs in blocking scheme
- After 1-1 matching
Practical Considerations

Question: Relative to what sample space?
- $A \times B$
- Pairs in blocking scheme
- After 1-1 matching

Each step presumably filters out a lot of low-weight pairs
False Non-Match Rate

- Difficult to determine as well as define
False Non-Match Rate

- Difficult to determine as well as define
- May as well try to estimate number of undiscovered matches in $A \times B$
False Non-Match Rate

- Difficult to determine as well as define
- May as well try to estimate number of undiscovered matches in $A \times B$
- Can try capture-recapture using *independent* blocking schemes
False Match Rate

Bellin-Rubin
False Match Rate

- Bellin-Rubin
- Larsen
False Match Rate

- Bellin-Rubin
- Larsen
- Larsen, Rubin, Winkler

Consider sample space without 1-1 matching

Consider sample space without 1-to-1 matching

Model as a mixture of 2 normal distributions (Box-Cox)

Consider sample space without 1-1 matching

Model as a mixture of 2 normal distributions (Box-Cox)

$M$ and $U$ must be well-separated and unimodal
Larsen, M.D. “Hierarchical Bayesian Record Linkage Theory,” Iowa State University, Statistics Department Technical Report
Larsen, M.D. “Hierarchical Bayesian Record Linkage Theory,” Iowa State University, Statistics Department Technical Report

Estimate error rates with 1-1 matching
Larsen, M.D. “Hierarchical Bayesian Record Linkage Theory,” Iowa State University, Statistics Department Technical Report

- Estimate error rates with 1-1 matching
- Complicated restrained optimization

- Estimate error rates with 1-1 matching
- Complicated restrained optimization
- Metropolis-Hastings procedure
Improved Parameter Estimates

Recall, if we had correct parameter values (and model), under Fellegi-Sunter, error rates are known.
Improved Parameter Estimates

- Recall, if we had correct parameter values (and model), under Fellegi-Sunter, error rates are known
- Improve parameter estimates using training data
Extended Likelihood Function

For unlabeled sample space $S$ and labeled training data set $T$, extended likelihood function

$$L = \left( \prod_{(a,b) \in S} \Pr(\gamma(a,b)) \right)^{1-\lambda} \left( \prod_{(a,b) \in T} \Pr(\gamma(a,b)) \right)^{\lambda}$$

for $0 \leq \lambda \leq 1$
Extended Likelihood Function

For unlabeled sample space $S$ and labeled training data set $T$, extended likelihood function

$$L = \left( \prod_{(a,b) \in S} \Pr(\gamma(a, b)) \right)^{1-\lambda} \left( \prod_{(a,b) \in T} \Pr(\gamma(a, b)) \right)^{\lambda}$$

for $0 \leq \lambda \leq 1$

Estimate using EM

$T$ is sample of pairs from the clerical review region that have been clerically reviewed

$T$ is sample of “pseudo-truth” data: pairs with sufficiently high or sufficiently low weight
Data Preparation

- Files must have matching fields of fixed length and location
Data Preparation

- Files must have matching fields of fixed length and location
- Matching fields are compared on a character by character basis
Data Preparation

- Files must have matching fields of fixed length and location
- Matching fields are compared on a character by character basis
- Unnecessary inconsistencies must be removed before matching is done
Basic Preparation

- Consistently encode categorical variables
Basic Preparation

- Consistently encode categorical variables
  - Sex, race
Basic Preparation

- Consistently encode categorical variables
  - Sex, race
  - Date, age
Basic Preparation

- Consistently encode categorical variables
  - Sex, race
  - Date, age
- Spelling standardization
Basic Preparation

- Consistently encode categorical variables
  - Sex, race
  - Date, age
- Spelling standardization
  - Titles: Dr, Dr., Doctor
Basic Preparation

- Consistently encode categorical variables
  - Sex, race
  - Date, age
- Spelling standardization
  - Titles: Dr, Dr., Doctor
  - Nicknames: Bill, William
Basic Preparation

- Consistently encode categorical variables
  - Sex, race
  - Date, age

- Spelling standardization
  - Titles: Dr, Dr., Doctor
  - Nicknames: Bill, William
  - Standard words: Co, Co., Cmpny, Company
Basic Preparation, Cont.

- Identify and parse components
Basic Preparation, Cont.

- Identify and parse components
  - Names: last, first
Basic Preparation, Cont.

- **Identify and parse components**
  - Names: last, first
  - Addresses: number, street, unit
Address Parsing

16 W Main ST APT 16
RR 2 BX 215
Fuller BLDG SUITE 405
14588 HWY 16 W
### Address Parsing

<table>
<thead>
<tr>
<th>Pre2</th>
<th>Hsnm</th>
<th>Stnm</th>
<th>RR</th>
<th>Box</th>
<th>Post1</th>
<th>Post2</th>
<th>Unit1</th>
<th>Unit2</th>
<th>Bldg</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>16</td>
<td>Main</td>
<td></td>
<td></td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>215</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>405</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fuller</td>
</tr>
<tr>
<td></td>
<td>14588</td>
<td>HWY 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>W</td>
</tr>
</tbody>
</table>
Business Lists

Much harder
Business Lists

- Much harder
- May have fewer comparison fields
Business Lists

- Much harder
- May have fewer comparison fields
  - Name
Business Lists

- Much harder
- May have fewer comparison fields
  - Name
  - Address
Business Lists

- Much harder
- May have fewer comparison fields
  - Name
  - Address
  - Phone
Business Lists

- Much harder
- May have fewer comparison fields
  - Name
  - Address
  - Phone
- These may not be unique
Business Lists

- Much harder
- May have fewer comparison fields
  - Name
  - Address
  - Phone
- These may not be unique
- May be difficult to parse
Example of Business Name Parsing

DR John J Smith MD
Smith DRY FRM
Smith & Son ENTP
Example of Business Name Parsing

<table>
<thead>
<tr>
<th>Pre</th>
<th>First</th>
<th>Mid</th>
<th>Last</th>
<th>Post1</th>
<th>Post2</th>
<th>Bus1</th>
<th>Bus2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>John</td>
<td>J</td>
<td>Smith</td>
<td>MD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Smith</td>
<td></td>
<td></td>
<td></td>
<td>DRY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Smith</td>
<td></td>
<td>Son</td>
<td>ENTP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Two Kinds of Standardizer

- Deterministic
Two Kinds of Standardizer

- Deterministic
- Rule based
Two Kinds of Standardizer

- Deterministic
  - Rule based
- Probabilistic
Two Kinds of Standardizer

- Deterministic
  - Rule based
- Probabilistic
  - Hidden Markov model
Rule-Based Standardizer

U.S. Census Bureau software
Rule-Based Standardizer

- U.S. Census Bureau software
- Based on extensive expert experience
Rule-Based Standardizer

- U.S. Census Bureau software
- Based on extensive expert experience
- Created for a specific sample space
Hidden Markov Standardizer

- Adaptable to different sample spaces
Hidden Markov Standardizer

- Adaptable to different sample spaces
- Based on training data

*The Australasian Data Mining Workshop.*


Hidden Markov Standardizer Reference


- FEBRL Project (Freely Extensible Biomedical Record Linkage)
Hidden Markov Model

Identify a finite number of hidden Markov states
Hidden Markov Model

- Identify a finite number of hidden Markov states
  - first, last1, last2, mi, prefix, suffix
Hidden Markov Model

- Identify a finite number of hidden Markov states
  - first, last1, last2, mi, prefix, suffix
- Use training data to assign transition probabilities from one state to the next
Hidden Markov Model

- Identify a finite number of hidden Markov states
  - first, last1, last2, mi, prefix, suffix
- Use training data to assign transition probabilities from one state to the next
- Use training data to assign probabilities for observations having given hidden state
Hidden Markov Model

- Identify a finite number of hidden Markov states
  - first, last1, last2, mi, prefix, suffix
- Use training data to assign transition probabilities from one state to the next
- Use training data to assign probabilities for observations having given hidden state
  - Look-up lists
Hidden Markov Model

- Identify a finite number of hidden Markov states
  - first, last1, last2, mi, prefix, suffix

- Use training data to assign transition probabilities from one state to the next

- Use training data to assign probabilities for observations having given hidden state
  - Look-up lists
  - Coded rules
Hidden Markov Model, Cont.

- Break object into component observations, assign them initial Markov states
Hidden Markov Model, Cont.

- Break object into component observations, assign them initial Markov states
  - “sir”, “mick”, “jagger”, “mbc”
Hidden Markov Model, Cont.

- Break object into component observations, assign them initial Markov states
  - “sir”, “mick”, “jagger”, “mba”
- Compute the highest probability sequence of hidden states for the given observations
Viterbi Algorithm

Not feasible to compute probabilities for all possible paths $O(n^l)$
Viterbi Algorithm

- Not feasible to compute probabilities for all possible paths $O(n^l)$
- Dynamic programming algorithm $O(nl)$
Viterbi Algorithm

- Not feasible to compute probabilities for all possible paths $O(n^l)$
- Dynamic programming algorithm $O(nl)$
- Each state is arrived at by the most probable subpath (Markov property)
Much more time is likely to be spent preparing the data than performing the record linkage
Standardization Summary

- Much more time is likely to be spent preparing the data than performing the record linkage.
- Records that fail to be standardized will probably fail to be matched.
Matching programs
Matching programs

Matcher
Matching programs
- Matcher
- Bigmatch
U.S. Census Bureau Software

- Matching programs
  - Matcher
  - Bigmatch
- Auxiliary programs
U.S. Census Bureau Software

- Matching programs
  - Matcher
  - Bigmatch

- Auxiliary programs
  - Counter
U.S. Census Bureau Software

- Matching programs
  - Matcher
  - Bigmatch
- Auxiliary programs
  - Counter
  - EM
Matching programs
- Matcher
- Bigmatch

Auxiliary programs
- Counter
- EM
- Standardizer
Matching Programs: Matcher

Matcher
Matching Programs: Matcher

Matcher

One-to-one matching
Matching Programs: Matcher

Matcher
- One-to-one matching
  - Files should not have duplicates
Matching Programs: Matcher

Matcher

- One-to-one matching
  - Files should not have duplicates
- Pre-sort files according to blocking scheme
Matching Programs: Matcher

- Matcher
  - One-to-one matching
    - Files should not have duplicates
  - Pre-sort files according to blocking scheme
  - Can re-run program on residual files
Matching Programs: Matcher

- Matcher
  - One-to-one matching
    - Files should not have duplicates
  - Pre-sort files according to blocking scheme
  - Can re-run program on residual files
    - Resort files according to new blocking scheme
Matching Programs: Bigmatch
Matching Programs: Bigmatch

- Bigmatch
- No one-to-one matching
Matching Programs: Bigmatch

- Bigmatch
  - No one-to-one matching
  - Can be used for deduplicating file
Matching Programs: Bigmatch

- Bigmatch
  - No one-to-one matching
    - Can be used for deduplicating file
  - Do not pre-sort files
Matching Programs: Bigmatch

- Bigmatch
  - No one-to-one matching
    - Can be used for deduplicating file
  - Do not pre-sort files
  - Can run several blocking schemes
Matching Programs: Bigmatch

- Bigmatch
  - No one-to-one matching
  - Can be used for deduplicating file
  - Do not pre-sort files
  - Can run several blocking schemes
  - Can match several files to one file
Matching Programs: Bigmatch

- Bigmatch
  - No one-to-one matching
    - Can be used for deduplicating file
  - Do not pre-sort files
  - Can run several blocking schemes
  - Can match several files to one file
  - One file must fit into memory
Auxiliary Programs: Counter

- Counter program
Auxiliary Programs: Counter

- Counter program
- Simplified matching program
Auxiliary Programs: Counter

- Counter program
  - Simplified matching program
  - Counts number of times each matching pattern occurs
Auxiliary Programs: Counter

- Counter program
  - Simplified matching program
  - Counts number of times each matching pattern occurs
  - String comparator has (high) cutoff
Auxiliary Programs: Counter

- Counter program
  - Simplified matching program
  - Counts number of times each matching pattern occurs
  - String comparator has (high) cutoff
  - Provides input for EM
Auxiliary Programs: EM

- EM algorithm program
Auxiliary Programs: EM

- EM algorithm program
  - Estimates probability parameters for given file and blocking scheme
Auxiliary Programs: EM

- EM algorithm program
  - Estimates probability parameters for given file and blocking scheme
  - Has 2-class and 3-class versions
Auxiliary Programs, Standardizer

Standardizer
Auxiliary Programs, Standardizer

- Standardizer
  - Standardizes names and addresses
Auxiliary Programs, Standardizer

- Standardizer
  - Standardizes names and addresses
  - Rule-based parsing