Paper CC-41

Implementing a Bayesian Approach to Record Linkage

Lynn Imel and Vincent Thomas Mule, Jr
U.S. Census Bureau

ABSTRACT

The Census Coverage Measurement survey-based program estimated household population coverage of the 2010 Decennial Census. Calculating coverage estimates required linking survey person data to census enumerations. For record linkage research, we applied a Bayesian Latent Class Models approach to both 2010 coverage survey data and simulated household data. This paper presents our use of Base SAS® to implement the Bayesian approach. It also discusses coding adaptations to handle changes including removing hard-coded variable names to allow for varying input parameters.

DISCLAIMER

This report is released to inform interested parties of ongoing research and to encourage discussion of work in progress. Any views expressed are those of the authors and not necessarily those of the U.S. Census Bureau.

INTRODUCTION

In general, record linkage methods use comparisons (agreement patterns) of common fields to define the match status of linked records from two or more files. A given record can be linked numerous times (many-to-many matches) or restricted to matching only one other record (one-to-one match). Researchers use the combined data in a variety of ways, such as producing coverage estimates or identifying duplicate records.

Larsen (2009) proposed a Bayesian record linkage application with many-to-many matches that pairs records from two files. The method builds on agreement patterns from two latent classes (matches and nonmatches) and makes the conditional independence assumption of comparison fields (variables common to both files, such as age). This approach does not allow parameters to vary by block (blocks are a group of linked pairs that agree on at least one variable, such as agreement by phone number). In this paper, we present the Bayesian approach proposed by Larsen, discuss implementing the method and show results. The implementation and results discussions include details on how the code developed during research.

BAYESIAN APPROACH TO RECORD LINKAGE

The Bayesian approaches presented in Larsen's 2009 paper link records from two files (A and B). A linked pair of records from files A and B are referred to as (a,b). Each linked pair has *k* comparison fields (name, age,…) with agreement levels (for example, agree/disagree) defined as $\gamma_k(a,b)$. The agreement pattern of the comparison fields is stored in a vector $(\gamma(a,b) = \{\gamma_1(a,b), \ldots\})$. A linked pair's match status is defined as I(a,b) = 1 for match and I(a,b) = 0 for nonmatch.

In this section, we present Larsen's Bayesian method (described in Section 3.1, *Bayesian Approach to Latent Class Record Linkage Models*) that we implemented in Base SAS. This approach models the probability of an agreement pattern ($Pr(\gamma)=Pr(\gamma|M)p_m+Pr(\gamma|U)p_u$) from two latent classes (matches and nonmatches), makes the conditional independence assumption of comparison fields and uses Gibbs sampling to simulate posterior distributions. It does not allow parameters to vary by block or force one-to-one matches.

The method is as follows.

A. Select initial values of unknown parameters (initial parameters were based on previous survey matching results).

B. Repeat the following steps until convergence:

1. For each linked pair of records, draw values for the match status ("I") independently from a Bernoulli distribution with

$$Pr(M|\gamma(a,b)) = \frac{p_m Pr(\gamma(a,b)|M)}{p_m Pr(\gamma(a,b)|M) + p_u Pr(\gamma(a,b)|U)}$$

where p_m= probability of match given match status
p_u = probability of nonmatch given match status
$Pr(\gamma(a,b)|M)$= probability of observing agreement pattern among matches
$Pr(\gamma(a,b)|U)$= probability of observing agreement pattern among nonmatches

2. Given match status, define new values for the probability of match given agreement pattern, $Pr(M|\gamma(a,b))$.

- Draw probability of match, p_m.

$$p_m|I \sim Beta\left(\propto_M + \sum_{(a,b)} I(a,b)\,, \beta_M + \sum_{(a,b)} \big(1 - I(a,b)\big)\right)$$

Set probability of nonmatch (p_u) to 1-p_m.

- Calculate probability of observing agreement pattern among matches, $Pr(\gamma(a,b)|M)$.

$$Pr(\gamma(a,b)|M) = \prod_k Pr(\gamma_k|M)^{\gamma_k} \left(1 - Pr(\gamma_k|M)\right)^{(1-\gamma_k)}$$

For every *k*th comparison field, draw the probability of agreement given match.

$$Pr(\gamma_k(a,b) = 1|M,I) \sim Beta\left(\propto_{Mk} + \sum_{(a,b)} I_{a,b}\gamma_k(a,b)\,, \beta_{Mk} + \sum_{(a,b)} I_{a,b}\big(1 - \gamma_k(a,b)\big)\right)$$

- Calculate probability of observing agreement pattern among nonmatches, $Pr(\gamma(a,b)|U)$.

$$Pr(\gamma(a,b)|U) = \prod_k Pr(\gamma_k|U)^{\gamma_k} \left(1 - Pr(\gamma_k|U)\right)^{(1-\gamma_k)}$$

For every *k*th comparison field, draw the probability of agreement given nonmatch.

$$Pr(\gamma_k(a,b) = 1|U,I) \sim Beta\left(\propto_{Uk} + \sum_{(a,b)} \big(1 - I_{a,b}\big)\gamma_k(a,b)\,, \beta_{Uk} + \sum_{(a,b)} \big(1 - I_{a,b}\big)\big(1 - \gamma_k(a,b)\big)\right)$$

IMPLEMENTATION PROCESS

We tackled programming the algorithm with a series of Base SAS statements and macro language. DATA steps and macros gave us desired control of programming. The control made it easier for us to test and evaluate the algorithm's detailed processing steps. For more on processing the algorithm, see the appendix.

The initial code processed simulated household data with specific characteristics. During our research, code requirements expanded to accommodate processing survey data and shorten run times. To meet the additional requirements, we made the following code modifications:

- Referencing Variable Names of Comparison Fields

  Comparison fields can vary for each application. To address the possible changes, we modified our code during research. When we initially designed the program, variable names of comparison fields were hard-coded throughout the algorithm. It made the code easy to follow but unable to process different comparison fields unless a user hard coded references. To avoid replacing references, we revised the hard-coded variable names with generic macro variables (&char1, &char2, …). Once the code processed generic references, we only had to modify one line of code to change comparison fields (%let varfields= …).

  Example A shows setup of the macro variables for three comparison fields: first name, last name and age. Users modify bolded text to process different comparison fields.

  Example A:

```
%macro setup;
   data _null_;
     %do j=1 %to &numvars;
       call symput("char&j","%scan(&varfields,&j)");
     %end;
   run;
%mend setup;

/* comparison fields - variable names */

   %let varfields=FNAME LNAME AGE;

/* number of comparison fields */

   %let numvars=%sysfunc(countw(&varfields));

%setup;
```

- Multiple Levels of Agreement

  The latent class approach Larsen describes in his 2009 paper is for comparison fields with two levels of agreement. In practice, our survey research data have at least three levels of agreement: agree, disagree and missing. To process more than two levels of agreement, we modified the algorithm's draws for observing agreement patterns from Beta to Dirichlet distributions. To accomplish this, we drew an independent Gamma variable for every possible agreement level then calculated the values' proportions. The resultant distribution of proportions is a Dirichlet distribution.

  Example B shows how to generate probabilities from a Dirichlet distribution for a comparison field with three levels of agreement: agree, disagree or missing.

  Example B:

```
/* Draw Gamma variables for 3 levels of agreement:
   agree(a1), disagree(a2) & missing(a3)*/

   a1=rand ('gamma',alpha1);
   a2=…
   a3=…
```

```
/* Produce Dirichlet Distribution */

   D1=a1/sum(of a1-a3);
   D2=…
   D3=…
```

- Processing Speed – Match Status

  Our survey-based research data consisted of Census Coverage Measurement (CCM) and census households. Each iteration of the code required over 750,000 independent Bernoulli draws to determine match status. If we use this type of technique in future census applications, the number of draws would be substantially larger (for example, matching 400,000 sample records to 300 million census records). Due to the size of future applications, we looked for ways to improve processing speed.

  One of the approaches we looked at to improve speed was reducing the number of records processed. We accomplished this by modifying determination of match status from a Bernoulli approach to Binomial. The first step of the Binomial approach is counting the number of links by unique agreement pattern.[1] Then, based on the number of links and probability of being a match, one draws the number of matches from a Binomial distribution. When we applied this approach, it reduced match status record processing from over 750,000 to around 6,000. For more on the Binomial approach, see Mule and Imel (2013).

  Example C shows how we drew match status from a Binomial distribution.

  Example C:

```
/*-------------------------------------------------------------------
  Binomial Draw of Match Status
    prob_I= probability of match given agreement pattern, Pr(M|γ(a,b))
    nlinks= # of links with unique agreement pattern
 -------------------------------------------------------------------*/

    I=rand('BINOMIAL',prob_I,nlinks);
```

RESULTS

SIMULATED HOUSEHOLDS

For our initial research, we simulated matching person records from two files that each had 100 households with 4 people. First, we generated many-to-many linked pairs (1,600 linked pairs; 100 blocks with 16 links) within each household (blocking variable). Links with the same person number (A1-B1,…) were assigned the same match probability (.9) and match status was drawn. All of the other links were designated nonmatches. In addition, we set agreement levels (agree or disagree) given match status for seven comparison variables: first name, last name, middle initial, month of birth, day of birth, age and sex.

To set each comparison variable's agreement level, we designated a probability of agreement, then drew agreement level from a Bernoulli distribution. We created two sets of the simulated data that differed by probabilities of agreement given match. The first scenario has probabilities of agreement greater than .9 for three comparison fields: first name, last name and age. These fields have lower probabilities of agreement (.75) in the second scenario. For each, we examined matches from our Bayesian implementation and the simulated households (truth). Table 1 shows the scenarios and results.

Table 1: Simulated Households

Simulated Households --- Scenario	Probability of Agreement Given Match							Matches	
	First Name	Last Name	Middle Initial	Month of Birth	Day of Birth	Age	Sex	Bayesian Approach*	True (Simulated Households)
1	**.95**	**.96**	.5	.6	.3	**.975**	.9	351.3	353
2	**.75**	**.75**	.5	.6	.3	**.750**	.9	359.9	354

*Estimate based on each iterations (after first 100) independent draw of matches.

[1] For example, links with two comparison fields each with three levels of agreement (agree, disagree, missing) can have up to nine unique combinations (3x3).

These results only indicate what can happen if probabilities of agreement given match (simulated truth) vary. In practice, we do not know true values and have more than two levels of agreement. Therefore, next we adapted the algorithm to process additional agreement levels and survey-based research data.

COVERAGE SURVEY

Our survey-based research data[2] consisted of CCM and census households that reported the same phone number. Blocking on households yielded over 60,000 blocks and many-to-many matching of the household person records resulted in over 750,000 links. Record linkage software, BigMatch, generated the initial many-to-many linked pairs of records. In addition to linking the records, BigMatch calculated an agreement score for each comparison variable (first name, last name, middle initial, month of birth, day of birth, age and sex). For first and last name, we used the agreement scores to form five comparison levels: exact agreement, strong partial agreement, weak partial agreement, disagree, and missing. We formed three levels of agreement (agree, disagree or missing) for all of the other comparison variables. For more on the BigMatch software, see Yancey (2007).

We compared the results of our Bayesian implementation to the CCM computer matching results. Our implementation of the Bayesian approach yielded an estimate of 199,112 matches[3] and CCM computer matching identified 203,196 matches[4]. In addition, to examine differences, we identified agreement patterns with differences of 100 or more matches. There are 17 patterns with differences of this size. Table 2 shows the agreement patterns with the largest differences.

Table 2: Matches by Agreement Patterns

Agreement Patterns							Matches		
First Name	**Last Name**	**Middle Initial**	**Month of Birth**	**Day of Birth**	**Age**	**Sex**	**Bayesian Using Mean***	**CCM Computer Matching**	**Difference (CCM-Bayesian)**
Exact	Exact	Missing	Disagree	Missing	Missing	Agree	268	1,080	812
Disagree	Exact	Disagree	Agree	Agree	Agree	Agree	1,022	531	-491

* Estimate based on the number of links and the probability of being a match (iterations after burnin).

The 17 patterns with differences of 100 or more matches fall into two groups: 1) more CCM computer matches than matches estimated by the Bayesian (using mean) approach and 2) more Bayesian (using mean) approach matches than CCM computer matches. There are 12 patterns in the more CCM computer matches group and five patterns in the more Bayesian approach matches group. The first row of Table 2 shows the largest difference (812) for the patterns in the group with more CCM computer matches and the last row of the table shows the largest difference (491) for the patterns in the group with more Bayesian approach matches. For more on applying Larsen's Bayesian approach, see Mule and Imel (2013).

CONCLUSION

Code used to explore research methods often must be adapted to deal with requirement changes. This paper presents how we handled this situation when implementing a Bayesian record linkage approach presented in Larsen (2009) and a few of the research results. Using Base SAS and macro processing, our code was adapted to handle varying inputs and processing speed requirements.

[2] The CCM conducted interviews, an operation referred to as Person Interview (PI), at sample-housing units in late summer of 2010. The interviews collected demographic data and information to determine a person's residence on Census Day (April 1, 2010). The person data, PI data, were matched to census enumerations. The matching process included two activities: 1) computer matching and 2) clerical review. Our survey-based research data are from computer matching.

[3] Estimate based on each iterations (after burnin) independent draws of matches. We ran the algorithm 1,100 times with the first 100 iterations as burnin.

[4] Matches are records identified as matches or possible matches; only a few of the records were possible matches.

REFERENCES

Larsen, M. (2009). "Record Linkage Modeling in Federal Statistical Database," Federal Committee on Statistical Methodology 2009 Research Conference.

Mule, V. and Imel, L. (2013). "Bayesian Record Linkage Models for Census Coverage Measurement Matching," *2013 Joint Statistical Meetings Proceedings*, Social Statistics Section. Alexandria, VA: American Statistical Association.

Yancey, W. (2007). "BigMatch: A Program for Extracting Probable Matches from a Large File," Research Report Series RRC2007/01, Statistical Research Division, U.S. Census Bureau.

ACKNOWLEDGEMENTS

The authors would like to thank Patrick Cantwell, Scott Konicki, Kathy McDonald-Johnson, Katherine Jenny Thompson and Colt Viehdorfer from the U.S. Census Bureau for their valuable contributions to this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Lynn Imel & Vincent Thomas Mule, Jr.
U.S. Census Bureau
4600 Silver Hill Rd
Washington DC 20233

lynn.m.imel@census.gov
vincent.t.mule@census.gov

APPENDIX

Bayesian Record Linkage – Iteration Process

In general, iterations of the Bayesian approach presented in this paper process through the following steps:

Step	Description/Code	Example
1. Draw	probability of match $p_m \mid I \sim Beta\left(\propto_M + \sum_{(a,b)} I(a,b)\ , \beta_M + \sum_{(a,b)}\left(1 - I(a,b)\right)\right)$ `prob_m=rand('BETA',alphaM,betaM);`	p_m =.8
	for k comparison fields, probability of agreement given match $Pr(\gamma_k(a,b) = 1 \mid M, I)$ $\sim Beta\left(\propto_{Mk} + \sum_{(a,b)} I_{a,b}\gamma_k(a,b)\ , \beta_{Mk} + \sum_{(a,b)} I_{a,b}\left(1 - \gamma_k(a,b)\right)\right)$ `prob_agree_m`$_k$`=rand('BETA',alphaM`$_k$`,betaM`$_k$`);`	$Pr(\gamma_k(a,b) = 1 \mid M, I)$ first name = .9 last name = .8 age = .5
	for k comparison fields, probability of agreement given nonmatch $Pr(\gamma_k(a,b) = 1 \mid U, I)$ $\sim Beta\left(\propto_{Uk} + \sum_{(a,b)} \left(1 - I_{a,b}\right)\gamma_k(a,b)\ , \beta_{Uk} + \sum_{(a,b)} \left(1 - I_{a,b}\right)\left(1 - \gamma_k(a,b)\right)\right)$ `prob_agree_u`$_k$`=rand('BETA',alphaU`$_k$`,betaU`$_k$`);`	$Pr(\gamma_k(a,b) = 1 \mid U, I)$ first name = .1 last name = .2 age = .2

2. Calculate	probability of observing agreement pattern
	<u>match</u>
	$Pr(\gamma(a,b) \mid M) = \prod_k Pr(\gamma_k \mid M)^{\gamma_k}\left(1 - Pr(\gamma_k \mid M)\right)^{(1-\gamma_k)}$
	`p_vect_m = (prob_agree_m`$_1$`)**(field`$_1$`)`
	`*(1-prob_agree_m`$_1$`)**(1-field`$_1$`)`
	`*(prob_agree_m`$_2$`)**(field`$_2$`)`
	`...`
	`*(1-prob_agree_m`$_k$`)**(1-field`$_k$`);`
	<u>nonmatch</u>
	$Pr(\gamma(a,b) \mid U) = \prod_k Pr(\gamma_k \mid U)^{\gamma_k}\left(1 - Pr(\gamma_k \mid U)\right)^{(1-\gamma_k)}$
	`p_vect_u = (prob_agree_u`$_1$`)**(field`$_1$`)`
	`*(1-prob_agree_u`$_1$`)**(1-field`$_1$`)`
	`*(prob_agree_u`$_2$`)**(field`$_2$`)`
	`...`
	`*(1-prob_agree_u`$_k$`)**(1-field`$_k$`);`
	where, field$_k$ = agreement (0-disagree or 1-agree) of kth comparison field
		If all comparison fields (first name, last name & age) agree:
		$Pr(\gamma(a,b) \mid M)$
		= (.9)(.8)(.5)
		= .36
		$Pr(\gamma(a,b) \mid U)$
		= (.1)(.2)(.2)
		= .004

	probability of match given agreement pattern
	$Pr(M \mid \gamma(a,b)) = \frac{p_m Pr(\gamma(a,b) \mid M)}{p_m Pr(\gamma(a,b) \mid M) + p_u Pr(\gamma(a,b) \mid U)}$
	`prob_I=(prob_m*p_vect_m)/`
	`(p_vect_m*prob_m+p_vect_u*(1-prob_m));`
		$Pr\left(M \middle\vert \gamma(a,b)\right)$
		.8(.36)
		= ----------
		.8(.36)+ .2(.004)
		= .997

APPENDIX

Bayesian Record Linkage – Iteration Process, continued

Step	Description/Code	Example
3. Draw	match status `I=rand('Bernoulli',prob_I);`	`I=rand('Bernoulli',.997);`
4. Update	for the next iteration, update parameters (α, β; match counts)	$\sum_{(a,b)} I(a,b)$ = # of matches $\sum_{(a,b)}(1 - I(a,b))$ = # of nonmatches …….