Missing Data Problems in Record Linkage-How to Find Links and Non-Links

Liz Lorenzi

October 19, 2013

Abstract

In many different applications, people want to link records corresponding to the same person or entity. Within these sources subjects may have differences in their names, birth dates, addresses, etc. Because of these discrepancies, often it is difficult to determine whether two records are a match. We can assess this relationship between records from different sources using similarity scores and different methodology to find links and non-links. To fill in the missing information of whether records are matches or not, we can use iterative algorithms. In this report, we will study the differences and effectiveness of two algorithms that can be used in record linkage, Expectation Maximization and an algorithm described by Larsen and Rubin (2001).

1 Introduction

Record linkage aims to find matches between pairs of records from separate data files to consolidate the information into a single data file. These data files may be subject to errors in their entries. A quick example is given in Table 1. We can see that the two files in List 1 and List 2 most likely are matches, but both have some sort of typo in the information recorded. Our goal is to decide whether records are still matches even with small errors in the input of the information.

Table 1: Examples of Records Clerically Reviewed

	First Name	Last Name	BY	BM	BD
List 1	Herbert	Shubert	1958	10	22
	Ursala	Martin	1963	6	8
List 2	Herbert	Shubert	1953	10	25
	Urstala	Martine	1963	6	8

This paper explores two algorithms that are used in record linkage to decide whether records are matches or not. They are methods that are used in latent variable problems where the information needed is unobservable. The Expectation Maximization algorithm is used in many different contexts, including determining the parameters in the mixture model that determines whether records are links or non-links (the term link will refer to when we predict the status of a pair of records). The methodology presented by Larsen and Rubin uses mixture models also to determine the status of a pair of records and then uses outside information to continue to update the algorithm. We can think of the Larsen and Rubin as a semi-Bayesian type approach, where we use outside information to help with our predictions. This paper will first address the methodology of both the EM and Larsen and Rubin approach, then move into the analysis performed to compare the algorithms.

2 Methods

This report aims to compare two different methods to see how adding additional information from prior knowledge can be useful when estimating missing values. The EM algorithm is a Frequentist approach that finds the maximum likelihood estimator of parameters. It is often used in addressing missing data in many different

applications. We use the algorithm in record linkage to find conditional probabilities for a Fellegi-Sunter record linkage model.

Our method begins by taking two sets of records and finding the similarity scores between them. This is done using Jaro-Winkler scores, which takes into account edits, transpositions, and omission of letters in strings. We then choose a cutoff value to convert the similarity scores into a pattern of agreement. We have 2^F possible agreement patterns, where f=$\{1...F\}$ representing the number of fields used in comparison. Once the data is in this form, we implement the two algorithms.

2.1 Mixture Models

Before continuing on the methodology of the two algorithms, we should address some underlying features of the analysis. Both of these methods rely on the use of mixture models. Mixture models are probability models that model subpopulations of an overall data's population or in other words, represents the data's overall density estimate with multiple distributions. An example to illustrate this idea is presented below in Figure 2.1. In the picture, the idea of an overall distribution being split into sub-distributions is displayed; we can see the bimodal feature of the data makes for two separate distributions.

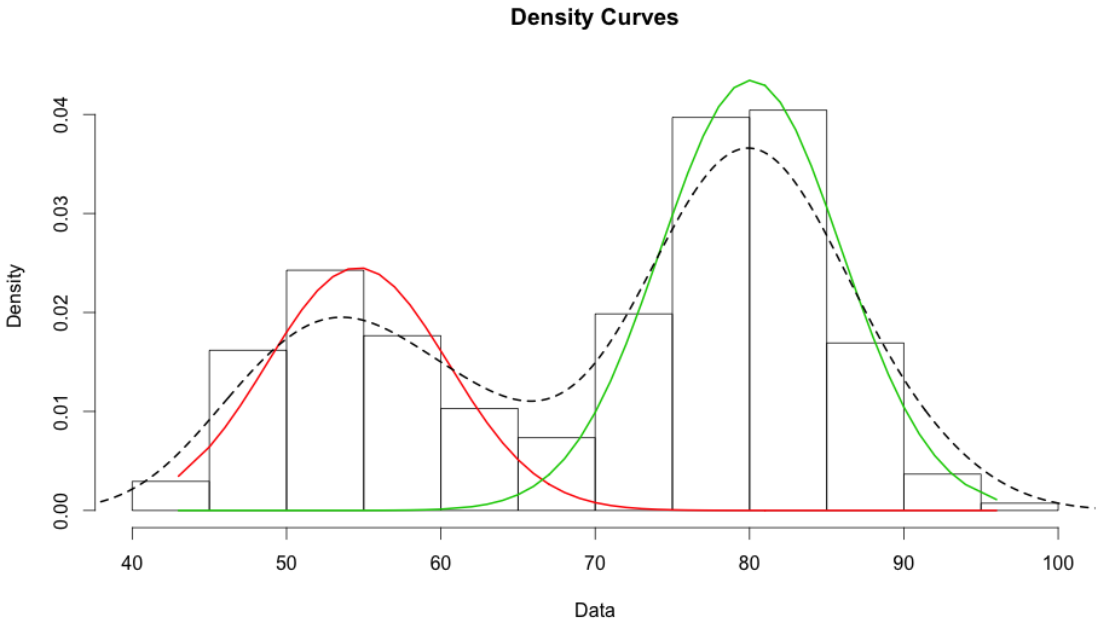

Figure 1: *Mixture model representation*

In our problem with record linkage, we can think about our data in this way. There tends to be a subpopulation that represents the records that match and a subpopulation of the records that do not match. This means that we can represent a distribution f as a mixture of K components, $f_1, f_2, ..., f_k$, more formally written below in equation (1) (Shalizi).

$$f(x) = \sum_{k=1}^{K} \lambda_k f_k(x; \theta_k) \tag{1}$$

λ_k represents the mixing weights, where $\sum_k \lambda_k = 1$, and θ_k represents the parameters of the distribution chosen to represent the k-mixture distributions. The distributions that make up the mixture models can be from the same parametric distribution, or they can be from different families of distributions.

2.2 Expectation Maximization

Expectation maximization (EM) is a tool used to find the maximum likelihood estimates of parameters in a probability model when this model depends on unobserved, latent variables. Specifically in record linkage, we use the algorithm to fill in the latent variable of whether two pairs of records are matches or non matches.

Specifically, we use EM to specify parameters necessary to implement the Fellegi-Sunter record linkage model. For this model, we have n*n pairwise comparisons between two sets of different records (with no duplicates) that we converted into agreement patterns across F fields (e.g. these may look like 10001, 00001 for 5 fields). Based on these agreement patterns, we can consider the probability that two records match by:

$$P(\gamma) = P(\gamma|r \in M)P(r \in M) + P(\gamma|r \in U)P(r \in U) \tag{2}$$

Where r is a pair of records, M represents Match and U, Unmatch, and γ is the agreement pattern (Fellegi and Sunter 1969). Using conditional independence, we have the marginal probabilities:

$$m = P((\gamma_1, \gamma_2, ..., \gamma_n)|r \in M) = \prod_{i=1}^{n} m_i^{\gamma_i}(1 - m_i)^{1-\gamma_i} \tag{3}$$

$$u = P((\gamma_1, \gamma_2, ..., \gamma_n)|r \in U) = \prod_{i=1}^{n} u_i^{\gamma_i}(1 - u_i)^{1-\gamma_i} \tag{4}$$

We begin my initializing the parameters needed to calculate the m_i and u_i probabilities as well as the P(M) which is computed by the number of pairs of records in set M (that are considered a match) divided by the total number record pairs, N. We use the equations described above to complete the data's log likelihood (5), then using our initial probabilities we calculate the function g for each pair and maximize the parameters. In equation (5), i represents the field and j is the pairwise set of records, where γ_i^j is 1 if field i has a similarity score larger than the cutoff assigned between record pair j, and 0 otherwise.

$$\hat{g}_j() = \frac{\hat{P}(m)\prod_{i=1}^{n}\hat{m}_i^{\gamma_i^j}(1-\hat{m}_i)^{1-\gamma_i^j}}{\hat{P}(m)\prod_{i=1}^{n}\hat{m}_i^{\gamma_i^j}(1-\hat{m}_i^{\gamma_i^j} + (1-\hat{P}(m))\prod_{i=1}^{n}\hat{u}_i^{\gamma_i^j}(1-\hat{u}_i)^{1-\gamma_i^j}} \tag{5}$$

Once we initialize the likelihood function, $\hat{g}_j()$, we can maximize the likelihood to retrieve new estimates for $\hat{m}_i$, $\hat{u}_i$ and $\hat{P}(m)$. This is the maximization step in the algorithm. We reiterate the steps described above until we notice little differences in our estimates for the parameters. For our purposes, I selected a cutoff of 0.001 in the difference between parameters from iteration t to $t+1$.

2.3 Larsen and Rubin Algorithm

In Larsen and Rubin (2001), a new method was proposed to maximize the likelihood of the record linkage mixture model and achieve better results using empirical information about the data and records being matched. The method they discuss is one that uses the marginal information of the data to find an appropriate mixture model, classify the record pairs as link, non-link, or clerical review. Then clerks review the pairs that are in the middle group (clerical review) and try to identify pairs as link or non-link. The model reruns on the updated information and continues until few records are able to be classified.

To implement the algorithm, we begin in steps similar to the EM algorithm described above. We can define an observation y_i from a finite mixture distribution with G classes as the probability density displayed in (6).

$$p(y_i|\pi,\theta) = \sum_{g=1}^{G}\pi_g p_g(y_i|\theta_g) \tag{6}$$

Let y_i represent a pair of records, π_g represent the probability of being in group, g, and $p_g(y_i|\theta_g)$ the density for that observation, and θ_g the parameters needed for the model (Larsen and Rubin 2001), where g represents the groups, match and non match. Similarly to the EM algorithm setup, we work off the basis of the Fellegi-Sunter record linkage model, where we look at the agreement patterns of pairs of

records. We then can think of the conditional probabilities for each pair as the probability of a having an agreement pattern, l, given that we are a match or non match. This set of probabilities make up the parameter, θ_g. Using Bayes theorem, we can rewrite equation (6) for unclassified data (without the knowledge of whether pairs of records are a match or not) as equation (7).

$$p(z_{ig} = 1|y_i, \pi, \theta) = \frac{\pi_g \pi_{l|g}}{\sum_{h=1}^{G} \pi_h \pi_{l|h}} \tag{7}$$

To find the maximum likelihood estimators for the likelihood equation (7), we use EM to find π_g, the probability of a match and the probability of a non match. To find the parameters, θ_i, for each pair of records, we use log-linear models and iterative proportional fitting to find new estimates for each cell of the table representing agreement patterns and match/non match.

The next step in the algorithm is to compare the maximum likelihood estimates to "semi-empirical" probabilities. The "semi-empirical" probabilities are derived using a combination of information from the data and prior-knowledge of the sets of records being linked. We then select the mixture model that minimizes equation (8).

$$\sum_{l=1}^{L} n_l [\tilde{\pi}_{lM} log(\tilde{\pi}_{lM}/\hat{\pi}_{lM}) + \tilde{\pi}_{lU} log(\tilde{\pi}_{lU}/\hat{\pi}_{lU})] \tag{8}$$

Once the probabilities are assigned to each pair of records using the optimal parameters above, we select the cases that become a link, and non-link, leaving the remaining in clerical review. Larsen and Rubin (2001) suggest possible cutoff values of 80% of the ratio of the size of the smaller of the two lists to the total number of pairs for the first value, and the second value being the model-estimated probability of being a link, choosing whichever is the smaller of the two. The links between these two cutoffs are sent to clerical review. Then the cases above the upper threshold are assigned a probability of a 1 and the pairs below the lower threshold are assigned a 0. We then refit the density for y and reiterate through this process until the number of matches the clerks find decreases to a small fraction of the number of cases reviewed.

3 Data

The data used for the analysis testing the methods outlined above is from the R package, RecordLinkage. The given data set in the package is called RLdata500 by Andreas Borg. Both of the methods above assume two sets of records are being observed with no duplicates in each set of records. To form a good subset to test methods on, we were given in class four lists that contain a random sample from the data with a somewhat even distribution for matches and non matches. We use the four sets of lists for the analysis below. List1 and list2 are both 100 records, lending to 100,000 pairwise comparisons, with 75 true matches. List 3 and list 4 each contain 75 records, with 5625 pairwise comparisons, with 60 true matches. We will not use the identifying column of the lists to test how well our algorithms classify the records as links and non-links. However, this will be used in comparing the accuracy of our methods.

Within the data there are 7 fields: Last name 1, Last name 2, First name 1, First name 2, Birth day, Birth month, and Birth year. Because of the high prevalence of missing values, we will not use Last name 2 or First name 2 in our analysis. We use list 3 and list 4 for both of the algorithms below.

4 Analysis

The initial set up of our analysis follow the method discussed by Fellegi and Sunter. This means we compare each record in the first list to every record in the second list, leaving us with n_1*n_2 possible pairwise comparisons (where n_i is the number of records in $list_i$).

Next, we measure the similarity between each pair of records. This can be done in a variety of ways such as exact matching, Jaro-Winkler, Levenschtein, and others. I chose to calculate similarity using Jaro-Winkler, which takes into account omissions, edits, and transpositions in strings. Once we have similarity scores for each field for each pair, we choose a cutoff value that changes the similarity score to a 0 or 1. We then look at the row of agreements and disagreements, and consider the pattern a cell, l. So for our 5 fields, this could be 00001 or 00101, denoting disagreement as a 0 and agreement as 1.

Once the data is in this form, we fit the mixture models for both possible algo-

rithms.

4.1 EM Results

We begin implementing EM by first giving it initial values for the parameters. I choose these arbitrarily being that EM is not sensitive to initial values. However, Fellegi and Sunter do suggest larger probabilities for m than u. Therefore, I choose 0.8 for all m's and 0.2 for all u's. I provide the initial value of P(M) to be 0.1. I then wrote two functions, the first to find the likelihood with the given estimates, and the second to check the change in the new parameter estimate and iterate again if it does not meet the threshold of 0.001.

The first function uses the estimates to find the likelihood of a match and non match given its agreement across fields then finding the overall probability of being a match given its full agreement pattern, which is $\hat{g}()$ shown in equation (5). However, this can be scaled down by instead finding the probability of a match given an agreement pattern, reducing the problem to calculate 2^L cases instead of n, where L is the number of unique agreement patterns and n being the total number of subjects. Once we have the probabilities $\hat{g}()$ for each case (filled in for each case if we did the reduced calculation), we calculate m^{t+1}, u^{t+1}, and $\hat{P}(m)^{t=1}$ through equations provided by Fellegi and Sunter.

The algorithm took 13 iterations to converge to the best estimates for the likelihood model. To determine which pairs should be links, non-links, or clerical review cases, I played around with different cutoff values. The probabilities that result from the final fit of $\hat{g}()$ were either all above 0.98 or below 0.02, with one case around 0.5. This gave me confidence to find cutoff values as U=0.9 and L=0.1, which contained the large tails of the resulting probabilities. The results are shown in Table 2. There was one case that resulted as a clerical review, however the rest of the link classifications were very accurate, with no false negatives or false positives.

Table 2: Final Results of EM

	True Matches	**False Matches**
Link	60	0
Non-Link	0	5564
Clerical Review	0	1

4.2 Larsen and Rubin Results

Due to time constraints, I modified the Larsen and Rubin algorithm described above for the following analysis. The conditional probabilities described in Section 2.3 in equation 6, denoted as θ_g, are calculated through model selection of different log-linear models that take into account different independence assumptions. This means that if conditional independence is not assumed for particular fields in the data, we can write different hierarchical models that take into account dependencies. We then use iterative proportional fitting to maximize the equations and find the MLEs. Instead of this method, I decided to use the conditional probabilities from list 1 and list 2 as my θ_g. I then perform the rest of the algorithm on list 3 and list 4. Because the data is from the same source, the probability that certain agreement patterns are matches and non-matches should be very similar between list 1/2 and list 3/4. I still recalculate the probability of a match through EM as described by Larsen and Rubin using only list 3 and list 4 likelihood calculations.

To first calculate the conditional probabilities for each agreement pattern given a match (θ_g), we have to use the identifying column of list 1 and list 2 to determine which are true matches and true non-matches. Through this we retrieve the following probabilities for the 29 agreement patterns that appear in our data in Table 3. These are the probabilities of an agreement pattern given that the pair is a match.

Table 3: Conditional probabilities of each agreement pattern given that the pairs are matches

00000	0	**11111**	1
00100	0	**00111**	0
00001	0	**11110**	0.875
00011	0	**01010**	0
00110	0	**10001**	0
00010	0	**11100**	0
10000	0	**10100**	0
01100	0	**01001**	0
00101	0	**10110**	0
11101	1	**01110**	0
11011	1	**10011**	0
10010	0	**10101**	0
11000	0		

After finding the likelihood for each pair, y_i, I find the MLE for π_M or the probability of a match. For our data, this probability is 0.010, which I selected when the difference between the previous value and the updated value is less than 0.01. After the first iteration with the likelihood equation using the probabilities shown above, 5565 record pairs are categorized as non-links, and 52 as links. This leaves 8 records in clerical review. I, serving as the clerk, then review the records in the clerical review section and decide whether to link these records or not. The records in clerical review, were all pretty obvious links. Below is an example of a few of the links. The first two entries I can confidently place as a match back into my algorithm, and the second two entries were a little less obvious. I placed a probability of a 1 on the first 7, and left the 8th (second set shown below) in clerical review, repeating the algorithm. Each pair in clerical review were of the pattern 11110. The new conditional probability of this cell with the updated information from clerical review is 0.984. With this update, the next iteration results in no observations in clerical review.

Table 4: Examples of Records Clerically Reviewed

	First Name	Last Name	BY	BM	BD
File 2 Example 1	Wolfgang	Becker	2007	4	75
File 3 Example 1	Wolfgang	Becker	2007	4	15
File 2 Example 2	Gerda	Richer	1934	2	12
File 3 Example 2	Gerda	Richter	1944	2	13

The final results show that the algorithm perfectly found the matches and non-matches. We can check this using the true IDs of the data. We have to realize that the modified way is expected to give us really strong results because we are using similar data to derive the estimates of the conditional probabilities for each agreement pattern. Table 5 presents the results, where we can see there are no false positives or false negatives.

Table 5: Final Results of Larsen and Rubin

	True Matches	**False Matches**
Link	60	0
Non-Link	0	5565

5 Discussion

Both of the algorithms explored in Section 4 performed very well. However, both had aspects that need to be discussed. First, we assumed for the EM algorithm that the fields are conditionally independent, meaning that each field for a single pair of records do not depend on any other fields. We can think of situations where this may not be true. For instance, maybe a certain person's ethnicity tend to have have longer names for both first and last names, meaning that a long name seen in the first name column would most likely also have a long name in the last name column. Despite this assumption, our algorithm ran very well, which is why I did not continue in exploring the dependent option for EM.

For Larsen and Rubin's algorithm, my implementation is altered greatly by using records from the same source that already have match status to find the parameters needed for the likelihood function used to predict probabilities for each case. However, the implementation I used is practical in many applications. For instance, the paper mentions the use of the algorithm for Census data. For Census data, we often have past information about records where records have been matched in the past and the match-status is known. If this is a case and we are sure that the sets of records refer to the same area, then using the method I implemented above is more practical and efficient method for finding links between records. Additionally, it performed very well.

It is difficult to compare the efficiency of both methods because one is completely automatic (EM) and the other updates through the use of human added information. At the same time, if both methods finds the same number of clerical review cases, Rubin and Larsen's method will be more efficient. Using the EM algorithm for record linkage relies on humans to decide on all of the clerical review cases. The Rubin and Larsen algorithm relies on clerks to update the information, and with the added information the algorithm can identify more of the cases in clerical review as either a link or non-link or can update past identifications that potentially were wrong with the new information. This can significantly decrease the number of cases the clerks need to review.

There are many aspects of both algorithms that I would like to further explore. First, for Larsen and Rubin, I would like to complete finding the maximum likelihood estimates for the parameters in the models. By using log-linear models with different independence and dependence assumptions, we add flexibility in finding the estimates that better fit the data. For the Larsen and Rubin example, I would also like to work in the empirical probabilities more. Because I used information from another similar data set for the actual estimations, I did not think adding additional information

known about the data was necessary. Performing these additional two steps would give me a much better view of the algorithm and how they intended for it to truly work. Additionally, I would like to test both algorithms on larger data sets to explore how each handles a large sample of data.

6 References

1. Thomas R. Belin Donald B. Rubin (1995) A Method for Calibrating False-Match Rates in Record Linkage, Journal of the American Statistical Association, 90:430, 694-707

2. Fellegi, Ivan P., and Alan B. Sunter. "A Theory for Record Linkage." Journal of American Statistical Association 64.328 (1969): 1183-210. Print.

3. Herzog, Thomas N., Fritz Scheuren, and William E. Winkler.Data Quality and Record Linkage Techniques. New York: Springer, 2007. Print.

4. Michael D Larsen Donald B Rubin (2001) Iterative Automated Record Linkage Using Mixture Models, Journal of the American Statistical Association, 96:453, 32-41, DOI: 10.1198/016214501750332956

5. Tanner, Martin A.Tools for Statistical Inference Methods for the Exploration of Posterior Distributions and Likelihood Functions. New York, NY: Springer New York, 1996. Print.

6. Shalizi, Cosma. "Mixture Models." 36-402 Notes.