Carnegie Mellon University
Dietrich College of Humanities and Social Sciences
Dissertation

Submitted in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy

Title: Linkage of Early 1900s Irish Census Records: Exploring the impact of household structure and crowdsourced labels

Presented by: Kayla Frisoli

Accepted by: Department of Statistics

Readers:

_____
Rebecca Nugent, Advisor

_____
Luiza Antonie

_____
William F. Eddy

_____
Robin Mejia

_____
Brendan Murphy

_____
Nynke Niezink

Approved by the Committee on Graduate Degrees:

_____
Richard Scheines, Dean Date

b

Carnegie Mellon University

Linkage of Early 1900s Irish Census Records: Exploring the impact of household structure and crowdsourced labels

A Dissertation Submitted to the Graduate School

in Partial Fulfillment of the Requirements for the degree

Doctor of Philosophy

in

Statistics

by

Kayla Frisoli

Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University

October 2020

Dedicated to my sister, Kendall.

Acknowledgements

I would first like to thank my collaborators and committee members for their help with this work. I greatly appreciate the support of my committee members: Luiza Antonie, Bill Eddy, Robin Mejia, Brendan Murphy, and Nynke Niezink. Thank you to Corey Emery for helping to create fun Ireland maps and Paul Rouse for the historical perspectives. Thank you to Ben LeRoy for being the ultimate `R Shiny` guru.

Thank you to my advisor Rebecca Nugent for being such a critical part of my PhD experience. I still remember when I was only a first year and Sam said that there was no better advisor for me than you. It took a while to realize it, but I finally came to see that he was right. You not only played the role as my dissertation advisor, but my mentor and fearless advocate. I will always be grateful for the opportunities you opened for me, and the experiences we had together. From electric scooters in Thessaloniki to taxi rides in Long Island and cold nights in Dublin, we've shared so many memories. Hopefully we'll keep biting off more than we can chew and somehow always pulling it off, for years to come.

Thank you to my first advisors Steve Fienberg, Sam Ventura, and Jared Murray. We sure did make trips to the Census Bureau filled with fun (as well as ice cream, cheesy bread, and wine). Thank you to Kirstin Early for making the long drives entertaining, engaging in deep Census Bureau elevator discussions, and helping to provide Yelp reviews for the veg options at Suitland's finest restaurants.

I couldn't have asked for a better department for my PhD studies and that's largely due to the amazing faculty, staff, and students. Thank you for making the department a positive, supportive place for students to thrive. Valerie Ventura, thank you for all of the laughs–your support helped me stick it out when I needed it most. Thank you to Peter Freeman and Michael Harding for your leadership with the DSI projects. Thank you to Chris Genovese, Brian Junker, Jay Kadane, Ale Rinaldo, and Chad Schafer for your advising and support. Laura Butler, Jess Paschke, Kira Bokalders, Mari Alice McShane, Heidi Sestrich, and Margie Smykla–thank you for everything. This thesis wouldn't have been possible without the computing support of Carl Skipper–thank you for your technical help at all hours of the night and your general willingness to always lend a helping hand.

The students in our department are hands-down the best around #beststudentsbestdepartment. Thank you to my cohort for making the first two years unforgettable: Ilmun (Dwight) Kim, Xiao Hui (Sherry)

Tai, Neil (Corneilius) Spencer, Alden (Brownies) Green, YJ (Wild Ride) Choe, Jaehyeok (Hawk) Shin, Mike (Stan) Stanley, Richard (Pennsylvania Plovert) Wang, and Daren (girls girls girls) Wang. Ilmun and Xiao Hui, thank you for the spontaneous trips, the make-up tutorials, the office dance videos, the seven-minute workouts, and our over-caffeinated discussions about cultural differences. I wouldn't have made it through without you two. Thank you to Neil for the Wendy's 4-by-4 before the 705 exam, the midnight beers on our walks home from the office, and always being down for a trip to Twisters. Thank you Alden for our (sometimes heated, but always fun) discussions, coffee and food tasting adventures, and outdoor explorations. Thank you to Nate, Joe, Octavio, Jackie, Yufei, Ivy, Robin, and Brendan. Michael (Vespers) Vespe, thank you for always being on my team. Collin Politsch, thank you for always having my back and for our Shadyside shenanigans. Purvasha, thank you for approaching everything enthusiastically and seeing the world in a positive light. Maria, thank you for forcing me to work out, for our nights watching The Handmaid's Tale, and being such an amazing friend. Thank you to Ben for tirelessly working to make everyone around you so happy. Christopher Peter Makris, you are truly such an amazing human being and I am so lucky to know you. Thank you to Kevin, Nil-Jana, Heejong, Tim, Nic, Matteo, Manjari, Ciaran, Riccardo, Theresa, Addison, Yue, Natalia, Alan, Beomjo, Pratik, Sasha, Shamindra, Wanshan, and Ron for the amazing memories. Thank you to Nic Kim and Mattia Ciollaro for the help navigating the job market. Thank you to Sam Adhikari, Maria Cuellar, Natalie Klein, Jerzy Wieczorek, and Francesca Matano for being such great role models. Thank you to the Random Walkers for giving me something to look forward to on random Tuesday nights at 9pm. Thank you to Abby Smith for partaking in lawn Olympics, advocating for wine walks, and generally for being such a positive light in this world. Amanda and Shannon, thank you for teaching me that tennis is played in games/sets/matches, for all of the Intramural championships, and for providing the ultimate sanity checks for everything I do. Shannon (Shanners) Gallgaher, thank you for the kite flying, the pumpkin picking, and for the reminders to plan my outfit when I'm stressing about a project. Mostly, thank you for your friendship which I am more grateful for than you know.

Thank you to my colleagues in the useR Meetup, the Pittsburgh Data Jam, and especially the Women in Statistics (WinS) Group. Hosting WiDS with you all has been a CMU highlight. Thank you to Amanda, Shannon, Xiaoyi, Maria, and Mikaela for being the best co-organizers. Thank you to Ernest Ng, Siena Duplan, and Michael Gethers for making my Salesforce internship such a positive one. I'd like to give a shout out to Gaucho, Driftwood, and Noodlehead, for giving me something delicious to look forward to during both the tough and celebratory times.

To my first grade teacher and very first role model, Kate Lynch–thank you. As you and Jenny always say, it's all because of you two. To my middle school math teachers Ms. Wasson and Mr. Benton–thank you. To my high school statistics teacher, Karen Davis, thank you for your friendship, support, and inspiration. Deepest thanks to my educators at UCLA: Professors Sanchez, Lew, Christou, Gould, Cochran, and Almohalwas. Juana, thank you for being my biggest cheerleader (or as you were recently called, the

ultimate “hype woman”) and advocate. I wouldn’t be where I am without you. Thank you to Barbara Wendelberger for inspiring me to go to graduate school and for being there for all of the big decisions.

To my first college roommates, Brittany and Maddy, thank you. Thank you for the study parties, the meatball subs, Vegas nights, football games, In-N-Out on my birthday, putting up with my shenanigans, and for coining “Dr. Friz”. Elliot, thank you for always holding me to the highest standards. Jordan, thank you for consistently making me laugh until I cry and calling me everyday even when I ignore you. Katie, thank you for being the ultimate GPhi sister and my all time go-to friend. Thank you to my UW SIBSlings for inspiring me to pursue graduate school and making Summer 2017 one of the best. To my “twin” Allie, thank you for being my math camp partner-in-crime and for being down to fit in a few hours of work even when we’re on vacation.

Thank you to my family. Aunt Monica, Uncle Dave, and Derek, thank you for believing in me and always providing me (and everyone you know) a warm place to stay and a glass of Chardonnay. Aunt Maria and Uncle Raul, thank you for showing me unconditional love and putting family before everything else. I’ll never forget our wild trip exploring (literally all of) the ’burgh. Thank you to the Hannas and the Loveladys for making Thanksgiving my favorite holiday. Bridget–I’m so grateful to be able to call my cousin one of my best friends; thank you for everything. Thank you the Halls, Aunt Dana, and Cole for your love and support. Thank you to Mommom for the countless games of Gin and shopping trips. Thank you to my mom for being my #1 fan, for supporting me in any and every way, for hanging out with me on Facetime while I cook in the kitchen, and for proof reading my thesis (sorry about that!). Thank you to my dad for always having the best advice and for imparting wisdom without realizing it, thank you for being someone I can turn to without judgement, regardless of the circumstance. To Kendall, for whom this thesis is dedicated, thank you for being you. Thank you for feeding me bread through the hole between our rooms that one time I got sent to my room, covering for me in high school, and taking the time to write the most heartfelt cards and leave the most supportive voicemails. Love you to the moon and back.

Abstract

Record linkage is the process of identifying records corresponding to unique entities across data sets. Linking individuals in historical data allows researchers to better characterize topics like population mobility, impact of local/national events, and generational changes. Historians in Ireland are currently interested in linking the recently released 1901 and 1911 census record databases. Like with many (historical) record linkage applications, there are challenges arising from the digitization of hand-written records, high frequencies of common names, and human mobility. Traditional methods struggle with these issues, and it is often acknowledged that specific sub-populations (e.g., women who change their names, individuals who move between census dates) are linked with lower accuracy. Additionally, these methods often consider only pairwise record comparisons without incorporating household or relationship information across records. Furthermore, development and assessment of supervised record linkage methodology often relies on labeled data sets with unknown label quality.

To help address these challenges, we designed a record linkage interface to study the impact of the human labeling process on the full record linkage pipeline. Via this interface, workers not only link records at the individual level but also at the household and within-household level, matching 1901 Ireland census records to their (potential) 1911 counterparts. In addition, we collect multiple instances for each label to assess label uncertainty. Our work capitalizes on this label collection process as well as known historical changes and the data's household structure. We find evidence that models incorporating this information better link hard-to-match populations.

Beyond linking the actual records and households, we collect information about how the labeler interacts with the interface (e.g., time spent, click patterns), providing rich information across labeler populations. Our approach was iteratively adapted to balance worker engagement, label quality, and monetary expenses. We find differences in downstream record linkage model performance based on changes in label generation and argue that it is critical to pay attention to these changes when labeling records or building models with pre-existing data. Data about the crowdsourced individual and household matches, the human labelers (from both CMU and Amazon MTurk), and the overall labeling process will be made publicly available. We

hope this data and our resulting insights prompt new areas of research within and beyond the record linkage community.

Contents

Chapter 1

Introduction

1.1 Early 20th Century Ireland

The early 20th century was a time of transition in Ireland, from religious reformation to the growth of cities [41]. Historians and scholars alike are particularly interested in changes in demographics, family structure, mobility, and the effects of world events across time. This is typically studied manually with the use of archival records and historical documents. However, recently the National Archives of Ireland transcribed, digitized, and publicly posted the country's original 1901 and 1911 census records as the respective 100 year embargoes expired (available at `www.census.nationalarchives.ie/` [42]). This process has, in turn, increased the accessibility of these records to the public and created a great opportunity for their study.

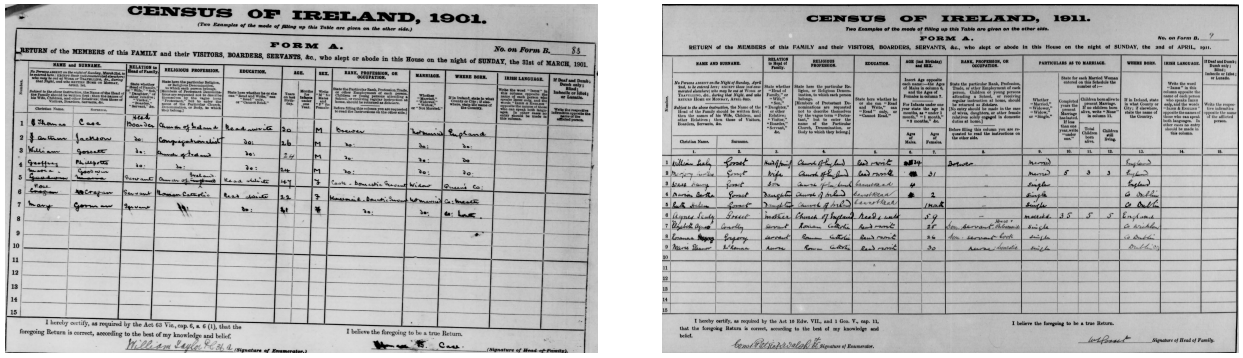


Figure 1.1: William Gossett, 1901 (left) and William Gosset, 1911 (right)

The Irish Census online database contains both the original records (Fig. 1.1) and their transcribed machine-readable counterparts (Table 1.1). In Fig. 1.1, we see the 1901 and 1911 census record for the household of William Gosset, the English Statistician best known for his development of the Student's t-distribution [44]. In the 1901 record (left) we see Gossett (misspelled) with other brewers and servants. In

1901

Surname	Forename	Age	Birthplace	Relation to Head	Occupation
Case	Thomas	30	England	Head of Family	Brewer
Arthur	Jackson	26	England	Boarder	Brewer
Gossett	**William**	**24**	**England**	**Boarder**	**Brewer**
Geoffray	Phillpotts	24	England	Boarder	Brewer
Goodwin	Maria	47	Queen's Co	Servant	Cook/Servant
Cregan	Rose	22	Co Meath	Servant	Servant
Gorman	Mary	41	Co Louth	Servant	Servant

1911

Surname	Forename	Age	Birthplace	Relation to Head	Occupation
Gosset	**William Sealy**	**34**	**England**	**Head of Family**	**Brewer**
Gosset	Marjory Surtees	31	England	Wife	
Gosset	Isaac Henry	4	England	Son	
Gosset	Marion Bertha	2	Co Dublin	Daughter	
Gosset	Ruth Helen		Co Dublin	Daughter	
Gosset	Agnes Sealey	59	England	Mother	
Connolly	Elizabeth Agnes	25	Co Wicklow	Servant	Servant/Parlourmaid
Gorgory	Rosanna	26	England	Co Dublin	Cook/Servant
McKenna	Marie Eleanor	30	Dublin City	Nurse	Nurse

Table 1.1: Original William Gosset(t) records transcribed and published online in the National Archives of Ireland.

the 1911 record (right) we see his immediate family, as well as servants and a nurse. An unusual feature of the Ireland Census is that household location is defined by a person's actual physical location on the stated census day at a pre-defined time. So households can and do include people who do not consistently live at that address (e.g., servants, visitors)*. In addition to name and relation to head of household, the record includes fields like religion, education, age, gender, occupation, and birthplace. The data are recorded at the household-level, meaning that individuals who were at the same physical location on the census day are recorded with the same household ID as everyone else at that location. The 1911 form (but not the 1901 form) also includes marriage status and information about children birthed. For every woman who has had a child, the record includes the total number of children birthed and the total number of children alive at the time of the census.

*This creates an additional challenge in that some individuals are double counted in multiple locations, and we can imagine that there may be systematic patterns to this duplication. Deduplication within a dataset is another area of record linkage and could be applied here as well[14]. But, we leave deduplication within the census years for future work.

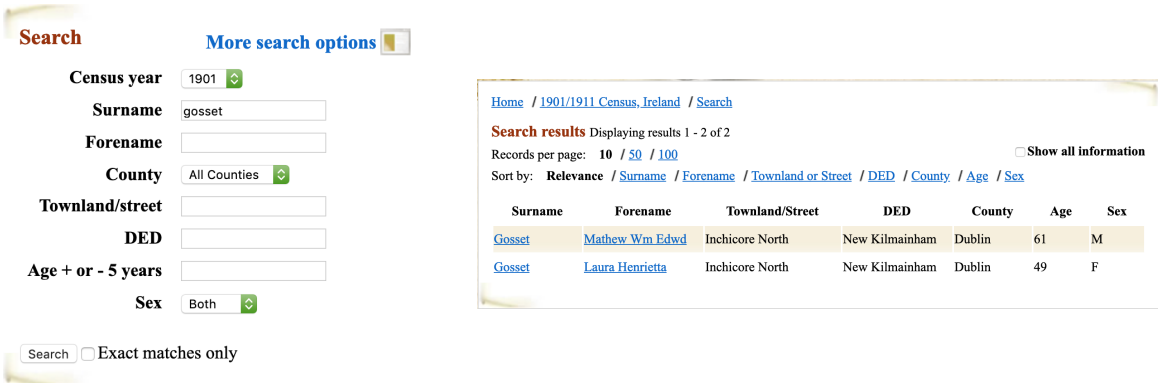


Figure 1.2: Ireland Census Database search box (left) and results (right). Notice that we do not find the William Gosset when we search for 'gosset' in 1901. This is because William Gosset's last name in 1901 was recorded as 'Gossett' with two 't's' instead of one.

Figure 1.3: Ireland Census Database search box (left) and results (right). William Gosset appears as the first record when we search for the term 'gosset' in 1911.

Irish historians currently utilize the existing digitized Irish Census Data base (Fig. 1.2) to extract and link information across the two census years, but there are limitations to this search mechanism [42]. For example, the use of exact-match only search parameters do not make allowances for "fuzzy matching" that would identify transcription or spelling errors. We see an example of this in Fig. 1.2 where the database does not find the correct William Gosset record as a possible matching 1901 record. His record is found when we search the same keyword in 1911 (Fig. 1.3). This limitation further complicates the linkage process if names or addresses have changed (e.g., (re)marriage or household moves). The search results are also not ranked, and historians tend to search one-by-one for potential links, which is an extremely time-consuming process [46].

The 1901 Irish population has 3.2 million people and this drops by about 80,000 over the next 10 years (the 1911 population is about 3.1 million). This drop in population is in part due to the combination of increased emigration out of Ireland and reduced immigration into Ireland. It is estimated that 399,065 Irish immigrated to the United States between 1901 and 1910 [56]. Another source cites as many as 4.5 million immigrants from Ireland to the US between the years of 1820 and 1930 [40]. Therefore, due to immigration and emigration in addition to births and deaths, we are aware that there will be 1901 records that do not have a matching 1911 record.

Our data is composed of counties (similar to United States counties) and the counties are further composed of electoral divisions (DEDs) which are composed of town streets. A population map of our data (from 1901) is shown in Fig. 1.4 (left) and a historic map of how County Carlow is divided is shown in the lower right of the image on the right.

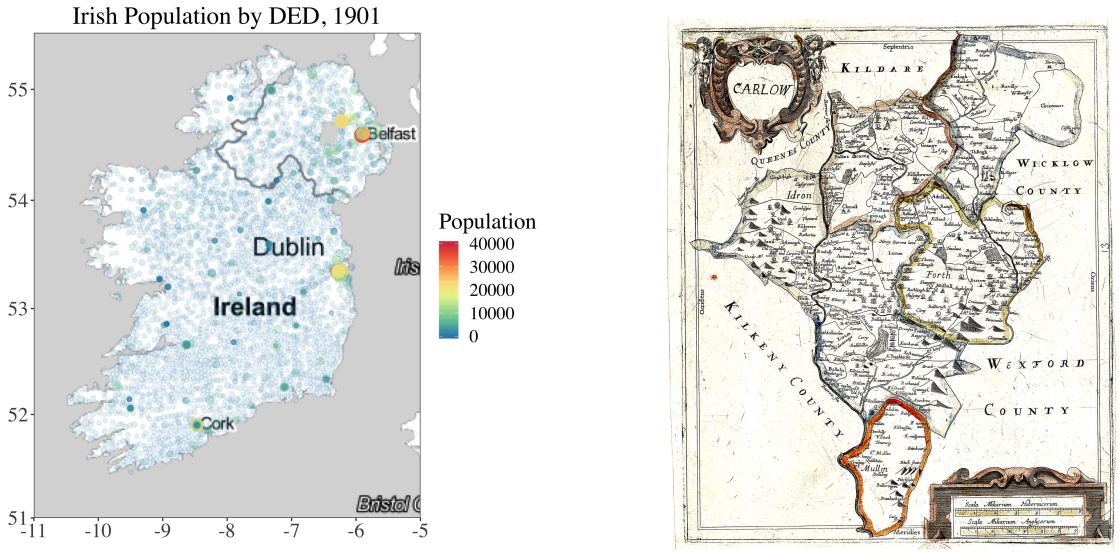


Figure 1.4: Ireland (left) is composed of counties which are composed of local administrative units called District Electoral Divisions (DED). County Carlow is shown on the right.

There are challenges that make linking the 1901 Irish Census to the 1911 Irish Census difficult, and the current data base is not set up for record linkage. Therefore, one thesis goal is to show the benefits of using statistical record linkage methodology to link the data.

1.1.1 Introduction to Record Linkage

Building upon the William Gosset example, we could compare the following two individuals (Table 1.1) and ask: Is the William Gosset in 1901 the same as William Sealey Gosset in 1911?

Year	Surname	Forename	Age	Birthplace	Relation to Head	Occupation
1901	Gossett	William	24	England	Boarder	Brewer
1911	Gosset	William Sealy	34	England	Head of Family	Brewer

From just the data above in Table 1.1, it seems likely that these individuals are the same person. But if we somehow knew that the name "William Gosset" was an extremely common name, or that being a brewer was a common occupation, we may not be so certain, especially given that there are no other common members across the two households (Table 1.1). Hand examining each record pair would be an impossible task. As such, we turn to statistical record linkage methodology to generalize matching records at a larger scale.

Record linkage is the process of identifying records corresponding to unique entities (e.g., individuals, companies) across data sets that do not have a unique identifier (e.g., Social Security Number, student ID). Often, record linkage approaches are classified as either deterministic or probabilistic. Deterministic approaches link records based on the number of exact matching features across the records[58]. Probabilistic record linkage, on the other hand, assigns weights to the feature comparisons and outputs a probability that the records match[20]. Probabilistic approaches typically outperform deterministic ones[58].

Often, statistical, probabilistic record linkage approaches follow a similar pattern, as shown in (Fig. 1.5)[13]. Records are first cleaned and standardized (if necessary) and then blocked[†] (if necessary) to reduce the number of comparisons. Pairs of records not in the same block are assumed to be non-matches. Then, pairs of records are compared within blocks using standard string/field similarity metrics (e.g., exact matching). Once pairs are compared, we can build statistical models to predict matching and non-matching pairs. If we wish to label pairs of records, we can set a cutoff and declare any pair that has a high enough predicted probability as a match. Otherwise, if we wish to provide a unique ID for each original record we will first need to resolve any transitivity issues (e.g., if A matches B and B matches C, then A should match C) if necessary by clustering, or some other heuristic approach [59]. We would finally assign unique identifiers to the individual records (that are shared across the two years).

Blocking

When looking for the (potential) 1911 matching record for a particular 1901 record, we could compare the 1901 record of interest to every 1911 record to find the one with the highest similarity. But, this is computationally very expensive (we'd need to make trillions of comparisons) and unnecessary (there is no need to compare "John Dalton" from Dublin to "Mary Murphy" from Limerick due to the dissimilarity of these records). Therefore, it is common in record linkage to block (partition) the data set into subsets of similar records and then only make comparisons within that subset[52]. For example, we may only

[†]Blocking is the process of partitioning data into similar subsets and only examining / comparing records within those subsets. Records across blocks are not compared. This is discussed further in the following Section (1.1.1)

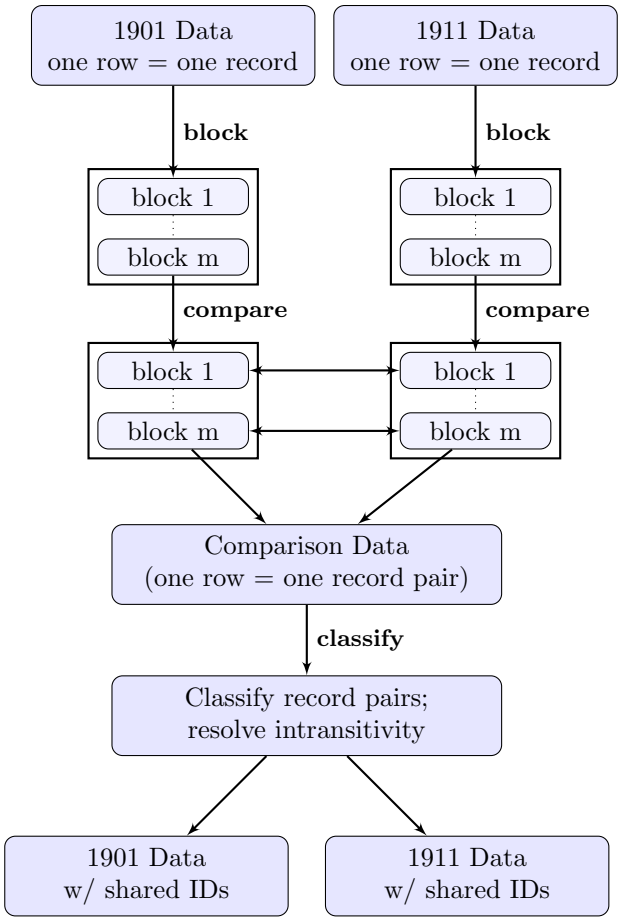


Figure 1.5: Flowchart of a common record linkage approach.

compare "John Dalton" from Dublin to men whose first names start with "J" and last names start with "D" within some geographic radius of Dublin. While there are many benefits to blocking, there are also downsides. Blocking introduces false negative errors (that can often not be recovered) because you cannot match individuals who were never compared.

Blocking is an often inevitable pre-processing step of record linkage. The blocks can be completely independent (e.g., we only compare individuals who have the same first letter of their first and last name) or can be made within non-independent passes (e.g., we compare individuals who match on first name *or* first letter of first and live in the same city). Blocks can also be learned via machine learning approaches[38][9]. There are numerous blocking approaches with benefits and costs to each[52]. In [7], blocks are creating using the Year of Birth and the Soundex code for Last Name. [8] uses first letter of first name and first letter of last name. Regardless of blocking approach, it is important to clearly state any blocking decisions[29].

Comparing individual records

When comparing two records to determine whether they refer to the same unique entity (i.e., match), we commonly analyze the similarity of the fields (e.g., first name, last name, age). Each field-pair (e.g., first name of record 1 to first name of record 2) is then assigned one, or multiple, metrics that represent the similarity between the entities. Common text string field similarities include the Jaro-Winkler (JW) [64] or phonetic similarity (e.g., Soundex [2]). Common numeric field similarities include absolute (numeric) difference (to capture numeric change) as well as string similarity metrics (to capture transcription error).

For example, a comparison of William Gossett 1901 vs. William Sealey Gosset 1911 might give:

Surname Jaro-Winkler	Surname Soundex	Forename Jaro-Winkler	Age Absolute Diff	Occupation Jaro-Winkler	**True Match**
0.97	1	0.9	10	1	?

There are many combinations of field similarities that we can construct in our comparison stage. Note that comparisons are made at the record-pair level, which is a limitation of common record linkage frameworks. Reliance on pairwise comparisons appear in both historical and non-historical record linkage, regardless of estimation approach (e.g., [7], [48]). We can consider comparing multiple (similar) individuals simultaneously or, multiple (similar) sets of individuals simultaneously (e.g., two households). More details on record and household similarity can be found in Section 5.2.

Modeling

If we were to know whether or not a record-pair comparison corresponds to a true match (shown below), then we could use supervised classification models to predict whether or not a future comparison corresponds to a match or non-match.

Surname Jaro-Winkler	Surname Soundex	Forename Jaro-Winkler	Age Absolute Diff	Occupation Jaro-Winkler	**True Match**
0.97	1	0.9	10	1	**1**

Assuming that the labels are of high enough quality, you can use your favorite supervised model (e.g., logistic regression, random forest, boosting) to link the two databases [31] [6]. However, these training labels can be expensive (in terms of both money and time) to obtain and difficult to create. Unsupervised classification models (e.g., Fellegi & Sunter [41]) assume that matched records have high similarity among a set of binary or discretized field comparison variables (e.g., exact match yes/no, $0.5 <$ Jaro-Winkler < 0.8). These methods also often assume conditional independence between the variables, which is limiting and

rarely met in practice (given that a pair is a true match, agreement on last name is likely not independent of matching on household location). This also creates issues when deriving string comparisons. For example, should we use Soundex or Jaro-Winkler to compare Forename? Using both would violate model assumptions and produce poor fitting models. There are advantages and disadvantages to both approaches and it is context-dependent / situational as to whether one would want to collect labels or fit unsupervised models.

We predominantly use supervised record linkage methodology, as early analysis showed these methods to be more promising in this application [22]. Since this data has never before been explored, there are no existing "truth" labels that link the two years. A second goal is to collect labeled data and study the label generation process, an aspect of record linkage that is seldom studied due to the fact that methodological papers often start with an existing labeled data set.

Final Linking / Decision Making

Regardless of how record similarities are determined, a decision needs to be made about whether two (or more) records refer to the same entity. Once we have a likelihood / probability of matching for pairs of records, we can set a probability cutoff and determine whether pairs are matches or not. Alternatively, we may want to assign unique IDs to the original records that are shared across the databases. As we assign IDs, we need to be wary of transitivity issues (Person 1 matches to Person 2 and Person 3, but Person 2 does not match to Person 3). To resolve this problem, we could cluster the records prior to assigning IDs, or incorporate more information (e.g., about household similarity) to determine which individuals should be linked. Details on this process can be found in Appendix A.1.

1.1.2 Challenges

In the context of historical Irish census records, traditional record linkage methods may struggle. Our records have limited (few, uninformative), non-standardized fields. Largely due to the time period, we find errors due to varying education levels of Irish citizens, changes in the format and style over the two years, as well as errors from the digitization of scanned, hand-written original records. In addition, we find a high frequency of common names (e.g., Mary, Murphy, Brendan) throughout Ireland in the early 1900s. Exploring the most common first name of each DED in 1901 and 1911, we see that the first names "Mary" and "Bridget" overwhelm the map seen in Fig. 1.6. Additionally, we see a striking geographic relationship among female first name popularity.

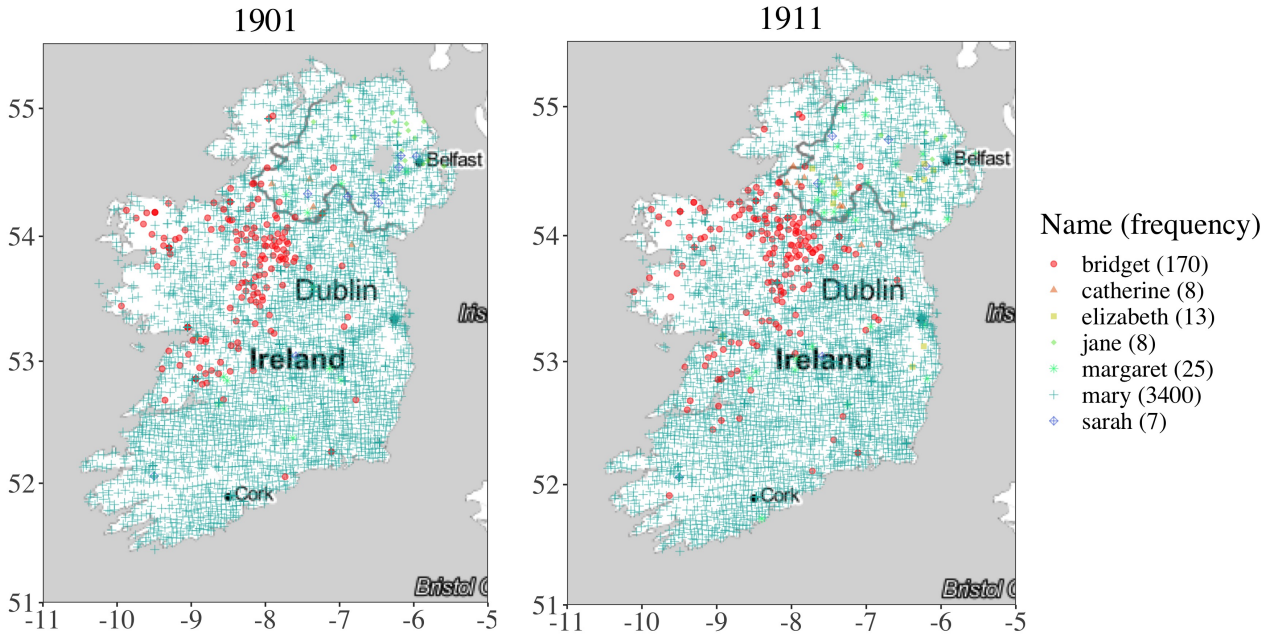


Figure 1.6: Most common first name in each DED across 1901 (left) and 1911 (right).

Because of these limitations, it makes sense to incorporate as much relevant information about the records into the modeling process as we can. As we saw in Fig. 1.1, the census is filled out by each household. Therefore, we inherently have additional information about a record originating from the other members listed on the form.

Often household information is provided in the original data, either directly or indirectly (Fig. 1.1). This additional structure is informative but often underutilized. Previous related work has been done within the historical record linkage framework. Antonie, Inwood, Lizotte, and Ross [8] employ a supervised record linkage approach to link 1871 and 1881 Canadian censuses. They utilize household information to secure confident matches but do not incorporate it within their modeling process. Group linkage methodology tries to more directly capitalize on existing group structure in the data [37] [43]. Li, Dong, Guo, Maurino, and Srivastava present a two step algorithm that first finds "pivots", or clusters, of similar records and then further merges records within pivots to find individual links. Fu, Christen, and Boot first link individuals with the expectation of some erroneous links, and then use existing household structure to refine and improve

1901:

	Surname	Forename	Age	Address	Relation	Religion	Birthplace
1.	Byrne	Mary	9	Portrushin Up	Daughter	Roman Catholic	Carlow
2.	Byrne	Mary	9	Portrushin Up	Daughter	R Catholic	Carlow

1911:

	Surname	Forename	Age	Address	Relation	Religion	Birthplace
3.	Byrne	Mary	20	Ticknock	Daughter	Roman Catholic	Carlow

1901 House 1

Forename	Age
John	45
Mary	33
Mary	9
John	6
James	0

1901 House 2

Forename	Age
Joseph	45
Anne	34
Anne	15
Joseph	11
Mary	9
⋮	⋮

1911 House 3

Forename	Age
John	55
Mary	42
Mary	20
John	17
Thomas	7
⋮	⋮

Table 1.2: There exist two (almost identical) Mary Byrne records in 1901 but only one similar record in 1911. It is unclear which of the first two (if any) records match with Mary from 1911. To better understand the situation, we can explore the individuals that were recorded on the same form as the Mary Byrnes. We now can see that Mary (1) more strongly matches to Mary (3) than Mary (2) due to the similar parents and sibling names.

the individual linkage results [23]. Although it seems intuitive (Table 1.2) that household or family structure would improve linkage results, given the complexities and variation in household structure and its availability across historical record linkage problems, the optimal way to incorporate this information is less clear.

A common problem in historical record linkage is how to handle expected field changes or differences [7] rather than typographical errors or common name variations (John vs Jon). If a woman marries, we expect her last name to change. If a family moves, we expect their location to change. The size and shape of a household unit will likely change over 10 years due to birth, death, and natural geographic movement. This is a challenging task without an obvious solution. Adopting the temporal record linkage approach of Hu, Wang, Vatsalan, and Christen may be a promising solution [32].

Our challenges do not end with data errors / lack of information at the record level. In consultation with Dr. Paul Rouse, a lecturer in the School of History at the University College Dublin, we discovered additional challenges due to historical policy reasons. During the Irish Land Act, land no longer had to be divided among all sons, but was given to the eldest son [16]. Younger sons often moved to cities like Belfast or Dublin to find work. So, searching for matches to younger sons (particularly in farming families) requires a larger geographic region (or a targeted search area based on typical job mobility trajectories) than for elder sons. In addition, social security benefits were introduced between 1901 and 1911 which provided incentive

to lie about age. A larger age difference than 10 years between 1901 and 1911 is not unexpected, especially among older individuals.

1.1.3 Thesis Outline

As introduced in this section, one focus of this thesis is to characterize the usefulness of household structure in the context of early 1900s Ireland historical record linkage. The other is to better understand the impact of record linkage labels and their generation on the full linkage pipeline, given that this data is currently completely unlabeled and we have a unique opportunity to start with the early linkage stages. To do this, we develop a linkage interface (see Chapter 2) that allows us to track and source the entire record linkage labeling pipeline. To crowdsource labels at faster speeds, we utilize the Amazon Mechanical Turk (MTurk) platorm. We discuss this process and it's challenges in Chapter 3. In Chapter 4 we discuss all aspects of our data. This includes labels we collect about individuals, households, and individuals within-households. It also includes metadata about the labeling process (including but not limited to label uncertainty, labeler quality, and labeler / interface interactions). In Chapter 5 we describe the process of comparing individual and household records and explore differences between metrics. We discuss supervised modeling approaches in Chapter 6 and report results for such models. We expand upon those models in Chapter 7 to explore differences based on the actual labeling process and our interface. We conclude in Chapter 8.

Chapter 2

Interface

To train our supervised models, we need ground truth data that links matching records from the 1901 census to the 1911 census. A labeler typically uses their best judgement to link individuals across data sets, and our goal is to train an algorithm to match this human judgement. However, there are currently no unique identifiers (e.g. a social security number) to link an individual from their 1901 record to their 1911 record due to the recent release of this data. This means the ground truth data needs to be collected 'by hand' if we want to utilize them. In order to facilitate this generation of ground truth data, we build an electronic interface that streamlines the process of hand-matching data. This interface is built in `R Shiny`, a web application that is available here: `https://link.stat.cmu.edu/Ireland`.

[[Kayla says: @RN Do we want to reference the DSAA paper here? Or more just introduce the DSAA and the ICDM papers in an intro section?]]

In this chapter we describe a novel record linkage labeling interface that attempts to address many of the record linkage challenges previously mentioned (Section 1.1.2). Due to these challenges, a successful labeling interface needs to have a flexible candidate identification process that does not solely rely on the similarity of individual record pairs. While this interface attempts to collect high quality labels, it also collects rich information about the labeling process. We can then study this process and potentially utilize process information in subsequent record linkage models. Increased emphasis on improving the data collection process (rather than focusing on methodology alone) during the initial labeling stages has the potential for a large impact on the quality of the final linked data. The label collection process is often treated as a black box and we seek to study it directly. Furthermore, commonly used public record linkage data sets like Krebsregister [19], RLData [10], German Cancer Registry [4], and [61] do not include any information on label quality, origin, or the collection process. [51] found that understanding and collecting information label had large impacts on downstream analyses. Therefore, the resulting data that we create with our interface

is unique and can have a large impact on the record linkage community. We discuss other interface benefits in Section 2.6.

Our novel record linkage interface:

- Provides additional group information to human labelers during the linkage process (e.g., household information in the Ireland census) and asks the labeler to link both the groups as well as individuals within groups;
- Improves the matching of individuals by finding those with expected field changes that would otherwise be missed using more traditional pairwise methods;
- Captures uncertainty in labels by having multiple individuals label a given record;
- Tracks labeler interface interactions (e.g., decisions / clicks) to collect information about the human labeling process;
- Supports the use of record linkage models that incorporate label uncertainty and human decision-making.

We will proceed in the following sections with a more detailed description of the labeling interface.

2.1 Interface Description

We present an interactive record linkage interface for collecting labeled individual, household, and individual-within-household records as well as information about how they were linked. We implement our interactive record linkage interface as an `R Shiny` application. The interface infrastructure can be broken up into three parts (depicted pictorially in Figure 2.1) that include:

(A) a flexible pre-processing phase dependent on the application/context that leverages and incorporates known structure about the data set at hand,

(B) a labeling interface that collects nested sets of human-labeled matches, and

(C) a back-end tracking of the human and computer interactions.

Application specific details of parts A, B, and C can be found in Subsections 2.2, 2.3, and 2.4, respectively. Additionally, this interface is highly adaptable and can include iterative feedback loops to enhance label collection and model performance (Subsection 2.5).

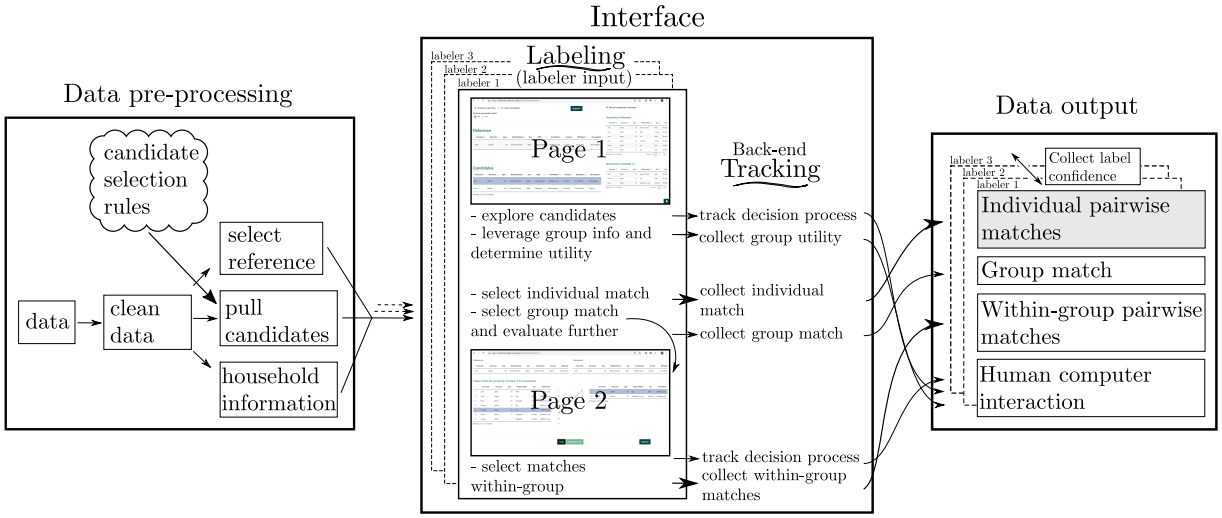


Figure 2.1: We present an overall diagram of the interface process. The interface contains 3 major steps: the data pre-processing, the physical labeling interface, and the data that is output at the end of the pipeline. Dependent on candidate section rules (see Table 2.1 Ireland-based examples), the data pre-processing step cleans and pulls the data that will be shown to the labeler. The labeling interface then presents the pre-processed data to labelers to collect matches for individuals, groups (in our case households), and individuals within groups. We also collect meta data about interactions between the labeler and the interface in order to capture the labeler's decision making process. The final component of our interface compiles and outputs the various data collected and tracked through the process.

2.2 Data Pre-Processing

To facilitate a more efficient labeling process, it is important to perform time-intensive computational steps before the labeler interacts with our interface. In our interface, these costly computational steps include identifying most-likely candidate matches and transforming them into a form which the labeler can easily understand.

2.2.1 Reference Selection

A reference record is a 1901 individual record that is shown to a labeler for linking. Our label collection was first focused on linking reference records within Ticknock, County Carlow due to its small population and rural nature (lending to a more homogeneous structure). We then progressively included more reference records from outside of Ticknock to capture individuals in the surrounding regions of County Carlow and County Wicklow. In order to capture geographic areas with a more diverse demographic structure, we collect labels from County Dublin and its surrounding areas, a much larger and densely populated area. Details about the specific data collected are provided in Section 4.1.

2.2.2 Candidate Selection

Given a reference record, we need to find a set of potential matching candidates. Record linkage often uses techniques (e.g., blocking [53]) to reduce the number of potential candidates that could match a given reference record. Similarly, before asking labelers (e.g., crowd-sourced or expert) to match individuals, one needs a process of narrowing down the set of potential candidates. We use rules that attempt to capture the true matches while accounting for both standard record linkage and application-specific expected field differences. This implicitly involves a trade-off between having strict rules that may exclude the true match and loose rules that might create so many candidates that they cause labeler fatigue.

Our approach is different than other record / candidate generating approaches, like Abramitzky (2014)'s iterative procedure [5] which increases the set of rules until a suitable number of candidates have been found or the strict guidelines from Collins and Wanamaker (2013) [15] that focus on finding only exact matches. Instead, we propose defining a set of both standard and application-specific rules that capture a wide net of potential candidates.

In our Ireland example, we use eight rules based on both common similarity heuristics and known historical and longitudinal information to generate potential candidates, as seen in Table 2.1. Rules ② and ③ allow for potential location changes among marriage-age men and women and last name changes among marriage-age women. Rules ④ through ⑧ allow for differences in one field, given that the rest of the record pair is extremely similar. In the interface architecture (Fig. 2.1, left block), we notate these rules as part of the data pre-processing phase and acknowledge their downstream influence. To help future researchers best leverage any collected labeled data and incorporate human subjectivity and biases found in the linking process into subsequent work, we strongly suggest that candidate selection rules be documented and made public.

Candidate Selection Rules

	Definition	
Common links	$\text{Distance}(\texttt{Location}_{1901}, \texttt{Location}_{1911}) \leq 8000\text{m}$, $\text{Exact}(\texttt{Gender}_{1901}, \texttt{Gender}_{1911}) == 1$, $\texttt{Age}_{1911} - \texttt{Age}_{1901} \in [5, 15]$, $\text{Jaro-Winkler}(\texttt{Surname}_{1901}, \texttt{Surname}_{1911}) \geq .8$, $\text{Jaro-Winkler}(\texttt{Forename}_{1901}, \texttt{Forename}_{1911}) \geq .8$	①
Married or mover men	$\text{Distance}(\texttt{Location}_{1901}, \texttt{Location}_{1911}) \leq 20000\text{m}$, $\text{Exact}(\texttt{Gender}_{1901}, \texttt{Gender}_{1911}) == 1$, $(\texttt{Age}_{1911} - \texttt{Age}_{1901}) \in [5, 15]$, $\text{Jaro-Winkler}(\texttt{Surname}_{1901}, \texttt{Surname}_{1911}) \geq .8$, $\text{Jaro-Winkler}(\texttt{Forename}_{1901}, \texttt{Forename}_{1911}) \geq .8$	②
Married or mover women	$\text{Distance}(\texttt{Location}_{1901}, \texttt{Location}_{1911}) \leq 20000\text{m}$, $\text{Exact}(\texttt{Gender}_{1901}, \texttt{Gender}_{1911}) == 1$, $(\texttt{Age}_{1911} - \texttt{Age}_{1901}) \in [5, 15]$, $\text{Jaro-Winkler}(\texttt{Forename}_{1901}, \texttt{Forename}_{1911}) \geq .8$	③
Strict rules	$\text{Distance}(\texttt{Location}_{1901}, \texttt{Location}_{1911}) \leq 2000\text{m}$, $\text{Exact}(\texttt{Gender}_{1901}, \texttt{Gender}_{1911}) == 1$, $(\texttt{Age}_{1911} - \texttt{Age}_{1901}) \in [5, 15]$, $\text{Jaro-Winkler}(\texttt{Surname}_{1901}, \texttt{Surname}_{1911}) \geq .95$, $\text{Jaro-Winkler}(\texttt{Forename}_{1901}, \texttt{Forename}_{1911}) \geq .95$	
Let distance differ	*Strict rules* $-\Big(\text{Distance}(\texttt{Location}_{1901}, \texttt{Location}_{1911}) \leq 2000\text{m}\Big)$ $+\Big(\text{Distance}(\texttt{Location}_{1901}, \texttt{Location}_{1911}) \leq 35000\text{m}\Big)$	④
Let gender differ	*Strict rules* $-\Big(\text{Exact}(\texttt{Gender}_{1901}, \texttt{Gender}_{1911}) == 1\Big)$	⑤
Let age differ	*Strict rules* $-\Big((\texttt{Age}_{1911} - \texttt{Age}_{19-1}) \in [5, 15]\Big)$	⑥
Let last name differ	Strict rules $-\Big(\text{Jaro-Winkler}(\texttt{Surname}_{1901}, \texttt{Surname}_{1911}) \geq .95\Big)$	⑦
Let first name differ	*Strict rules* $-\Big(\text{Jaro-Winkler}\ (\texttt{Forename}_{1901}, \texttt{Forename}_{1911}) \geq .95\Big)$	⑧

Table 2.1: We notate the rules that define the set of potential 1911 candidates for each 1901 reference record in our Ireland census application. Note that "Jaro-Winkler" stands for the standard Jaro-Winkler edit distance similarity[34]. The Jaro-Winkler similarity was used due to its common use in practice, but other similarity metrics could also be used to generate rules. Rule ① captures common record linkage candidates. Rules ② and ③ attempt to capture links among individuals who likely got married or moved between the two years. For each of rules ④-⑧, we start with the *strict rule* and then allow for one of the field similarities to be violated (e.g. ⑤ allows for the gender to differ between the two records).

2.3 Labeling Interface

Once the pre-processing is complete, we address the steps a labeler will take to match records. On the first page, one reference record and a set of candidate records is presented to the labeler along with individual and household (group) information about these records. The labeler is asked to determine if there are any matches between the reference and candidate records. If a labeler indicates there is a match, they are then presented with a second page that asks them to determine whether there are additional matching records from the households. The labeling interface is shown in the middle section of the interface architecture (Fig. 2.1).

Page 1

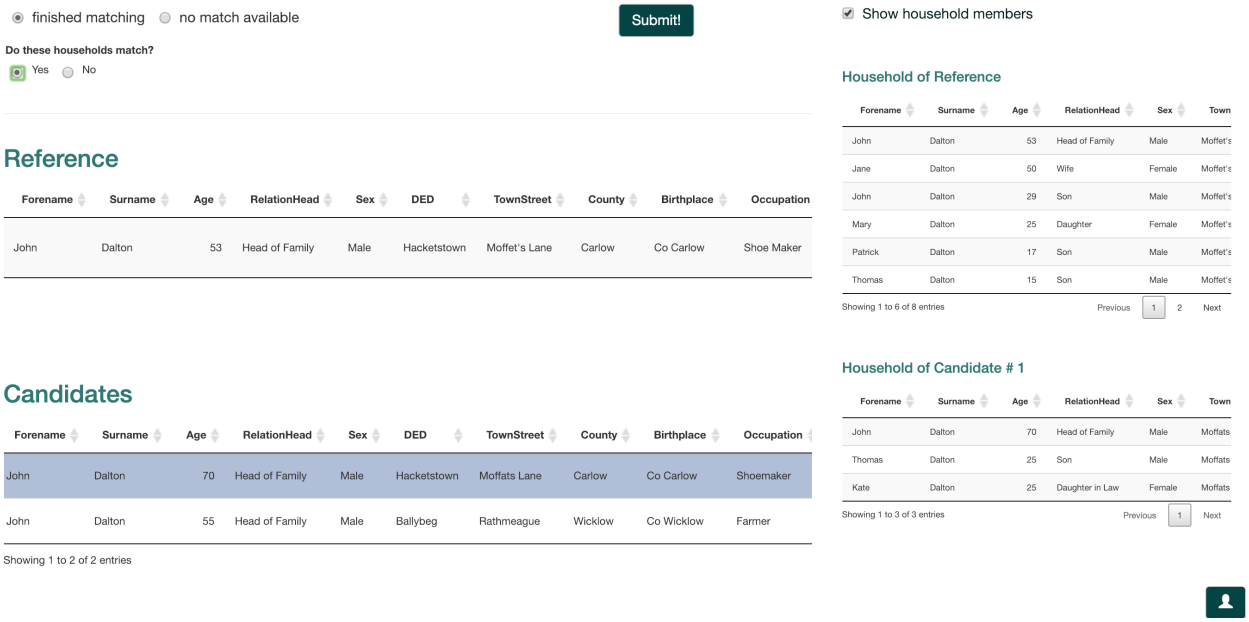


Figure 2.2: An example screen shot of Page 1 of the labeling interface. In this example only two candidates were similar enough to be included as potential matches. In the upper right hand corner the labeler selected to see the household members of "Candidate #1", shown below the reference record's household. In the upper left hand corner the interface collects whether there was a matching candidate and if the two households match (there are additional household members that should be linked).

Figure 2.2 presents an example of the first page of the interface. We ask labelers to provide a link (if one exists) between a reference record and a set of candidate records using our platform, which allows for intelligent interaction with the available data. We allow the user to explore all potential candidates further by reordering candidates by any record field, informally allowing the user to "re-weight" the usefulness of each of these columns. Additionally, and more importantly, the user can choose to view group membership and examine records from the reference and candidate households. From page 1 we store matches for individuals

(i.e., did the reference and the candidate match?) and households (i.e., did any individuals in the reference's household match to any individuals in the candidate's household?).

Page 2

Once the labeler determines whether a matching candidate exists at the individual level, they then decide whether or not the groups also match (i.e., at least one more pair of individuals across groups match). If the households are labeled as a match, the labeler moves to Page 2 (Fig. 2.3), and links matching individuals between households (stored as a within-group match). This second step captures a slightly different conditional probability than before, moving from $\mathbb{P}(\text{match} \mid \text{individual similarity})$ to $\mathbb{P}(\text{match} \mid \text{groups match, individual similarity})$. One would expect, on average, that the latter probability would be higher for a given similarity vector. If you know that the reference record's household contains matching individuals to the candidate's household, you might be more likely to match the reference to the candidate.

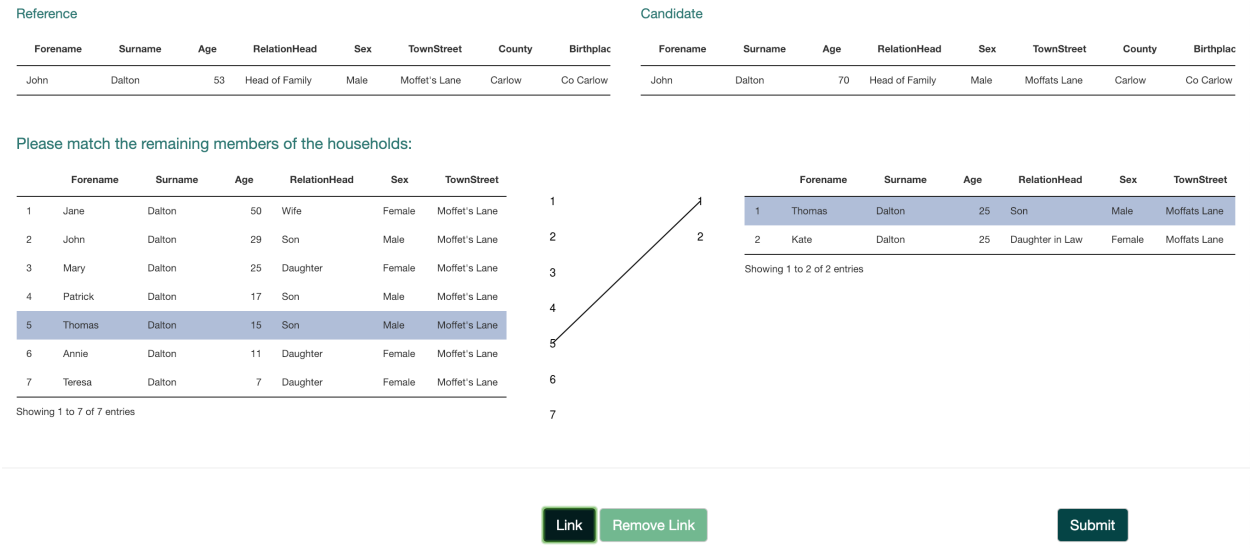


Figure 2.3: An example screen shot of Page 2 of the labeling interface. This page appears if the labeler believed that the households matched when examining them on Page 1 of the interface. One additional individual (Thomas Dalton) seems to belong to both households (and as such we see a link between the 5th individual in the first household and the 1st individual in the second household). It is not shown here, but in the last phase of data collection, we allow labelers to say that the household actually doesn't match and "fix" an incorrect house link.

One of the greatest advantages of linking additional individuals across households is that we can capture individual matches that might not be found via the candidate selection process (e.g., due to low similarity scores, name changes, typographical/transcription errors). One example of such a link is described later in

Variable Name	Description
`reference`	reference record
`candidates`	candidate chosen
`labeler ID`	optional ID of labeler
`household_useful`	was the household info useful? (boolean)
`household_match`	did the household match? (boolean)
`household_match_mistake`	did the labeler say the household matched on page 1 but change their mind on page 2? (boolean)
`household_click`	was the household information viewed? (boolean)
`clicks`	the order in which the labeler clicked to view potential candidates and their household information (e.g. `2_1_3_2`)
`label_source`	was the label from page 1 (individual page) or page 2 (within-household page)?
`time_on_page`	time spent on either page 1 or page 2

Table 2.2: A subset of the data that is tracked while the labeler interacts with the interface.

Section 2.6.2. Another advantage is the sheer increase in number of labels we receive (relative to time spent labeling); we discuss this in Section 2.6.1.

2.4 Back-End Tracking and Label Confidence

Throughout the linking process, we track how the labeler interacts with the interface. Specifically, we capture information on how the user explores the potential candidates, the speed at which they do so, and if they find the group information to be useful etc. (see Table 2.2 for a subset of the tracked information). The right panel of the 'Labeling Interface' box of Fig. 2.1 indicates the role of tracking within the overall framework. This allows us to study the utility of group information, how to collect higher quality links, and user responses to interface features. Beyond recording individual labeler decisions, we crowdsource multiple labels per reference record to capture label uncertainty. The idea of collecting repeated labels within crowdsourcing has been shown to be beneficial to overall label quality [51]. Furthermore, [63] found that error rates of the label decrease exponentially with more labels. This uncertainty can be incorporated into downstream models and interface adaptations. This idea is reflected in Fig. 2.1's global process with three tabs representing the three hypothetical labelers of a given reference record.

2.5 Adapting and Extending the Interface

Our labeling interface is designed to improve record linkage methodology and the quality of linked data sets through 1) better and more informative data collection and 2) an understanding of the labeling process. Each step of the record linkage process has downstream impacts and we additionally seek to understand /

quantify those impacts. Reference selection criterion affects which data the labelers and models see; candidate selection rules prohibit labelers from matching records outside of the pre-selected candidates; the interface's visual presentation affects the consistency and robustness of labeler conclusions; overall data quality affects what models we build and how they perform. Figure 2.4 embodies how the proposed interface fits into record linkage approaches and naturally adapts to include feedback from record linkage models.

Each of the feedback loops reflects potential updates to the process based on model performance or patterns in the collected data. As an example, the current interface infrastructure was expanded as a result of early back-end data collection showing high utility and impact of incorporating group (household) information in the linkage process. Preliminary findings relative to labeler uncertainty (discussed in Section C.1) and known benefits of incorporating active learning in crowdsourcing tasks [39] suggest additional dynamic features such as adaptively choosing the number of times a record is labeled and / or which record to show a particular user. Application-specific adaptations can also be included, as necessary. We discuss iterative changes that we've made to the interface as we've collected more data and received user feedback in Section 3.3.

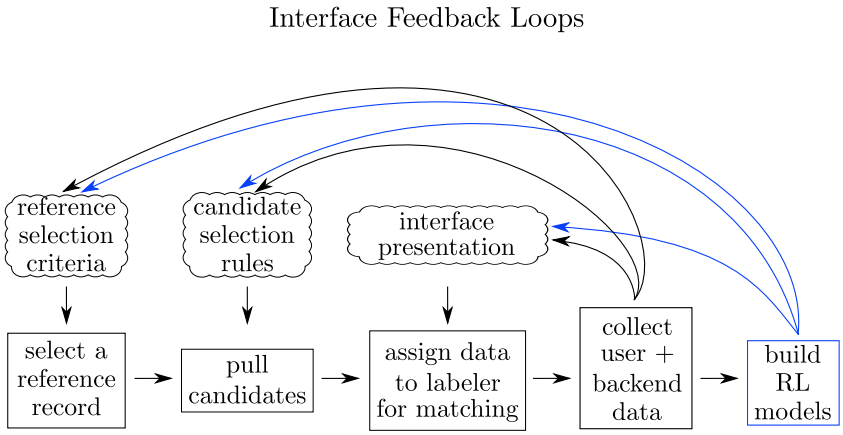


Figure 2.4: Potential for feedback / updates to the record linkage interface process. The square blocks represent the process of a reference record and the cloud bubbles represent "developer" decisions that affect the overall process.

Although this thesis focuses on the application of Irish Census records, the `R Shiny` interface can easily be adapted to other problems via two context-specific decisions and a data formatting requirement. First, the user needs to define the selection method for which reference records are shown to which labeler. Second, the user needs to define the candidate selection rules (see Subsection 2.2 for our candidate selection process). The last requirement is to flag the column that represents the group membership (in our case, the household ID). An adaptation of this interface has been used by the Center for Statistics and Applications in Forensic Evidence (CSAFE) at Carnegie Mellon University to label and link dark web seller accounts [54] [55].

2.6 Interface Benefits

A large product of this thesis is the unique data that we produce via the linkage interface. These data are applicable not only to record linkage but also to communities like human computer interaction (HCI) and crowdsourcing. These benefits are discussed further in the data and conclusion Chapters (4 and 8). In this section, we present some of the other benefits of using the interface for record linkage. Note that these benefits were found while linking a subset of the Irish Census records from 1901 and 1911, and may not necessarily be extrapolated to other applications. We assess the usefulness of including household information in the linkage process and how the interface can find links that may otherwise be missed.

2.6.1 Usefulness of Household Information

The first version of the interface (including only Page 1, which is shown in Fig. 2.2) asked labelers whether the household information provided in the right panel of the interface was useful in matching a candidate to the reference record. Overwhelmingly, 73.1% (71.3, 74.8)% of records had labelers who deemed this information useful. A visual representation of this data is shown in Fig. 2.5. Our first round labelers (from CMU) informed us that they found this information useful because the addition of (or lack of) other matching within-household pairs strengthened their certainty of a potential reference / candidate match.

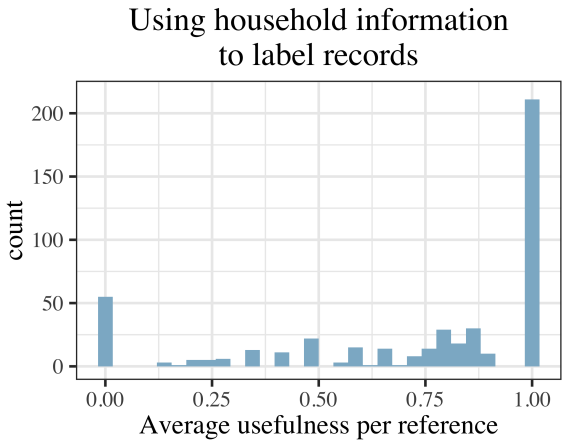


Figure 2.5: Household information usefulness. We indicate indicates the proportion of times users found the household information useful by reference record (1 = 100%); in general, users overwhelmingly found household information to be beneficial. In this dataset of 338 records from 1901 and 313 records from 1911, we have 105,118 pairs of records with only 204 matches.

To capitalize on the fact that labelers were already spending time examining the households, we decided to ask the labeler to input whether the households matched. Furthermore, we asked them whether any additional individuals should be linked across households. Once we changed the interface to collect this additional information (using Page 2, shown in Fig. 2.3), we found that 62.2% (58.6, 65.8)% of the time the

household of the reference record matched the household of the matching candidate. If there were only one additional matching household-household pair for every individual link, this would suggest 62% more records were able to be captured, greatly increasing our total number of labeled matches for little additional work. But, we know that labelers often find multiple matching members (e.g., an entire family) across households, giving 62% as a lower bound. In fact, in Table 2.3 we see that we actually get more than a 200% increase in the number of records by including the second page of the interface (and a 100% increase in the number of unique records).

	Individual (page 1)	Household (page 2)	Overall
Label instances	695	1638	2333
Unique references	632	1053	1391

Table 2.3: A summary of the data collected for the second CMU round of crowdsourcing. We show the number of reference records examined from both Page 1 and 2 of the labeling interface as well as the number of unique records. Unique records are applicable because reference records can be analyzed by multiple labelers both on Page 1 as well as on Page 2 of the interface.

Please note that all conclusions from this subsection were drawn from labels collected at CMU (but the results hold for data collected via Amazon MTurk–which we discuss in future sections).

2.6.2 Finding Unique Record Pair Matches

The linakge interface not only captures more links faster, but also helps to find unique and often commonly mislabeled matches.

We can see one example of such a record in Table 2.4. In 1901 we see the record for Thomas Neil (age 9) and in 1911 we see the record for Ths Kehoe (age 19), located in a different city. Based solely on the individual records (ignoring any household information), most algorithms and human labelers would not link the two together based on the dissimilarity of the record fields. But, it is highly likely that they are the same person once we examine the complete households. Table 2.4 shows the other memebers in the Kehoe / Neil households. We observed that a labeler using our interface was able to link "Thomas" and "Ths" on Page 2 of the interface after having linked Julia Neil, Thomas' mother, on Page 1.

This is just one example of a labeled match that we were able to collect via the interface, that would otherwise not be linked together. By collecting more accurate and diverse training data we are able to focus on building models that also find these hard-to-match links.

1901

Surname	Forename	Age	Location	Relation to Head
Neil	Thomas	60	Ticknock	House head
⋆Neil	⋆Julia	⋆55	⋆Ticknock	⋆Wife
Kehoe	Patrick	14	Ticknock	Son
Kehoe	Garrett	11	Ticknock	Son
Neil	**Thomas**	**9**	**Ticknock**	**Son**

1911

Surname	Forename	Age	Location	Relation to Head
⋆Neill	⋆Julia	⋆64	⋆Rathvilly	⋆House head
Kehoe	Patrick	24	Rathvilly	Son
Kehoe	Garret	21	Rathvilly	Son
Kehoe	**Ths**	**19**	**Rathvilly**	**Son**

Table 2.4: Originally a labeler linked the two Julia Neil records (denoted with a star ⋆). This linkage allowed a labeler to link Thomas Neil (1901) to Ths Kehoe (1911) even though these records were dissimilar and not an obvious match without the household information. Note that all other available fields (e.g., birthplace, religion) were consistent across all members of both households.

Chapter 3

Crowdsourcing Using Amazon Mechanical Turk

Amazon Mechanical Turk (`https://www.mturk.com/`) (MTurk) is an online platform that facilitates the crowdsourcing of work from around the world. [11] found that crowdsourced data is just as good as or better than in-person data collection. These findings have been corroborated by others like [28] and therefore we feel confident using a crowdsourcing platform like Amazon Mechanical Turk to label Irish Census Records. MTurk connects "requesters" with "workers". The requester (us, in this case) sets up a "task" for the workers. Tasks are released in batches, according to the specifications set by the requester. Workers can preview all available tasks and accept ones that they want to complete. We, the requesters, are notified once a worker submits our task. We pay a fee to both the worker and Amazon, and all payments are facilitated via the Mechanical Turk platform. More details on specific payments are in Section 3.1.1. After a batch is completed, the requester receives very limited information on the workers and the metadata regarding their work. We can then choose to either approve or reject the submitted task. Details on how we make these decisions can be found in Section 3.2. We can also block workers from completing any future tasks we make, which we do when we find a worker misusing the interface. During the Mechanical Turk crowdsourcing process, we receive feedback from workers. Using this feedback and interface tracking data, we make changes to the interface and the task setup in an attempt to receive higher quality labels from happier workers. Feedback and interface changes are documented in Section 3.3.

3.1 Setup

Projects are created using the "Create" feature. Amazon MTurk has customizable templates that requesters can use to design our tasks. They have multiple built-in templates for tasks like image classification, natural

language processing, and data collection. The requester is told that "You can use any HTML, CSS, or JavaScript to customize your layout". But, I had already built my label collection interface in `R Shiny`, and hosted it on an `R Shiny` server. Converting this interface to meet Amazon's requirements would have been time intensive and created issues consolidating and streamlining the data we collect. Additionally, when we checked with MTurk support staff in early 2020, it was impossible to have a worker complete multiple labels within a batch. Due to the nature of our interface, it did not seem an efficent use of ours or the worker's time to work one label at a time. For all of these reasons, we found it best to use the "Survey Link" template, shown in Figure 3.1 and send the workers to our website instead of embedding our interface within the MTurk platform.

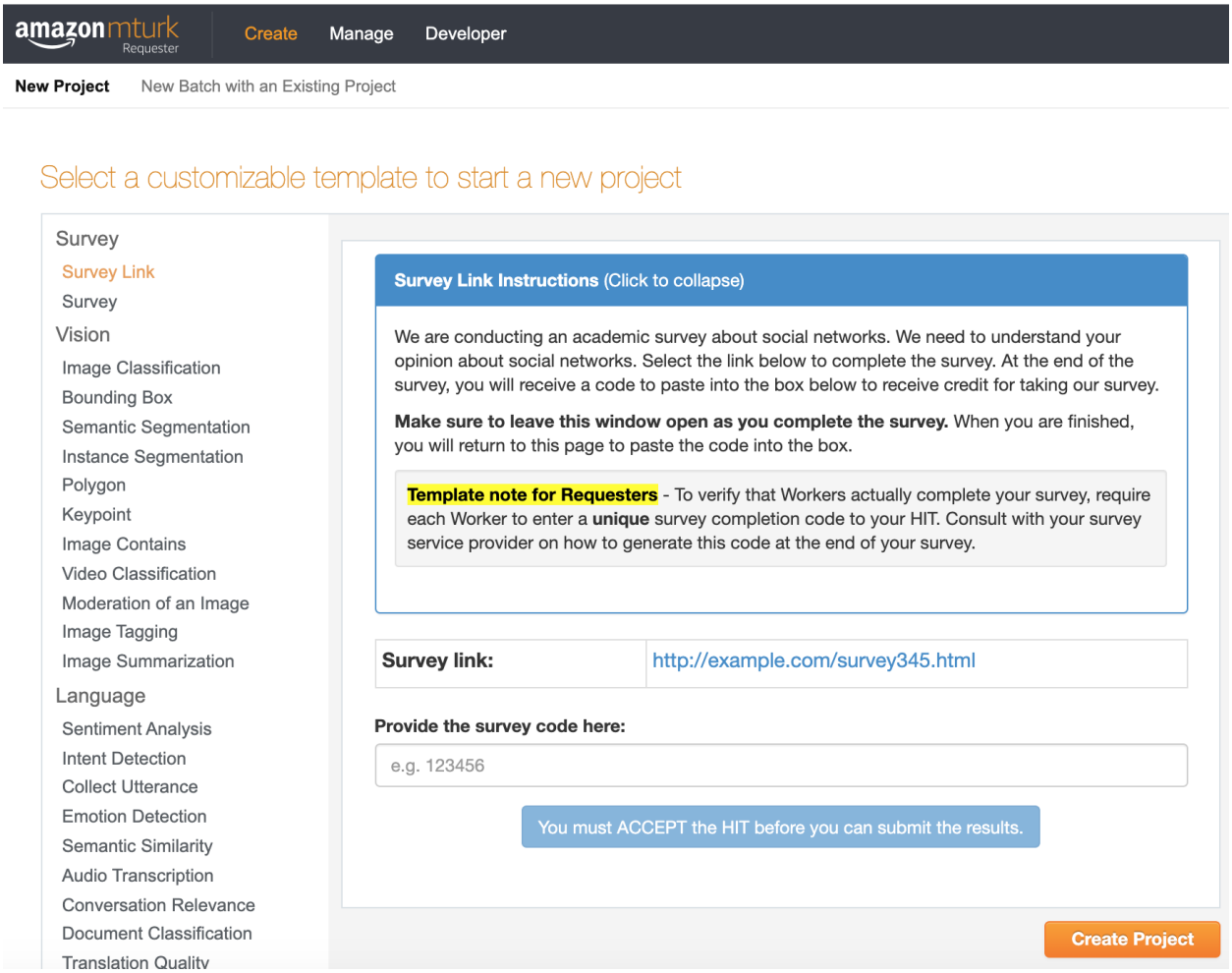


Figure 3.1: Example of an Amazon MTurk survey link template.

Once you click the orange "Create Project" button at the bottom of the template (shown in Fig. 3.1), you are sent to a new page (Fig. 3.2) to fill out information about your batch. In Fig. 3.2 we see the details

needed to create a new project, which we will discuss in the following subsections. In the upper portion, you can see that we must first provide a title, description, and keywords so that workers can search for our task.

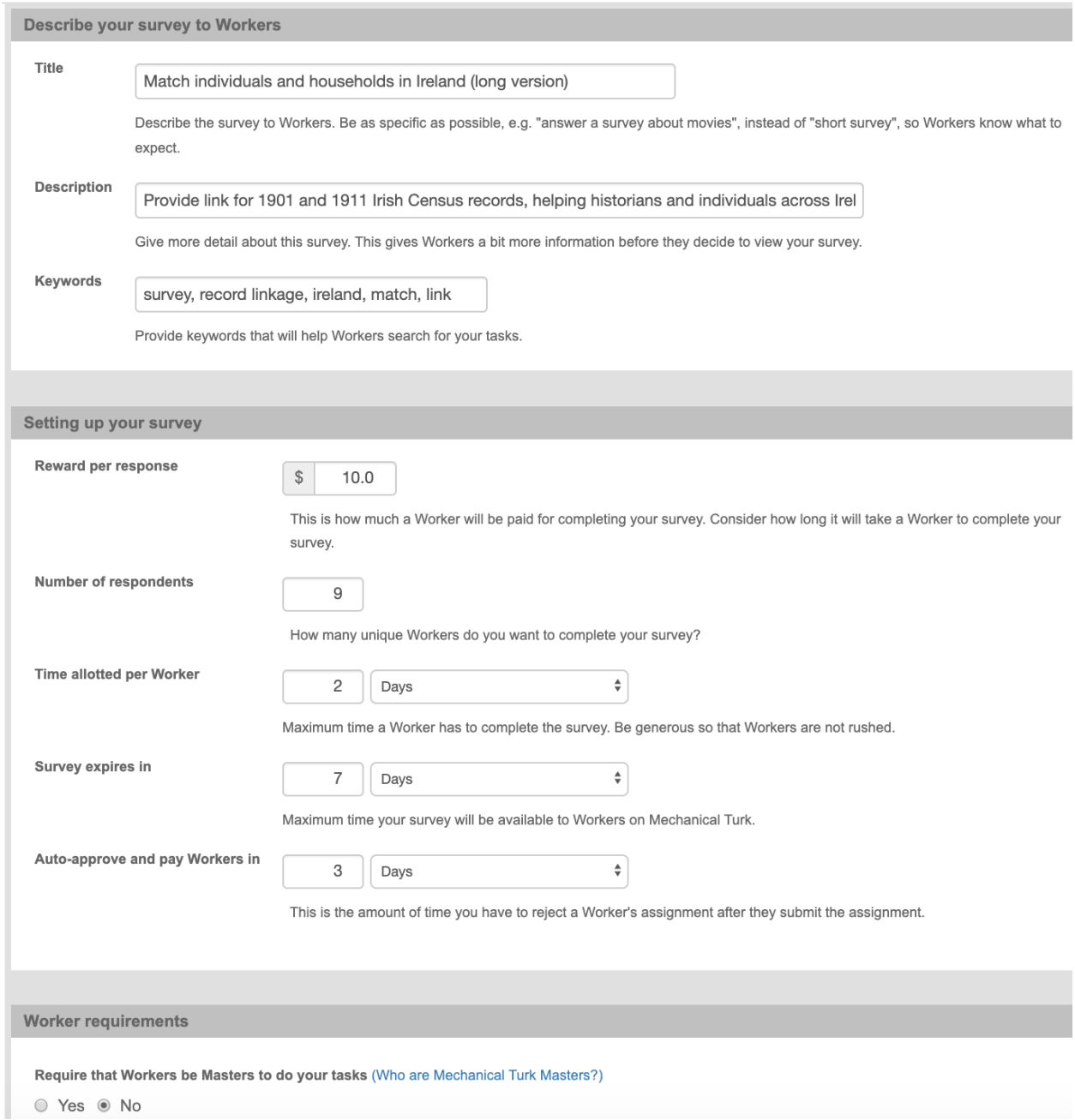


Figure 3.2: Template of the specifications required for an MTurk batch.

The remaining features of the template require a longer explanation, so we will detail those responses in the subsections below. Note that we made use of Reddit among other MTurk tracking platforms to understand common worker concerns and requester mistakes[3].

3.1.1 Payment and Number of Workers

In the second portion of the project form, we determine the quantity of workers we need and the amount of money we will pay them in USD. We determined to pay workers $0.10 per label (a submit on either Page 1 or Page 2) which we thought was competitive given the time a task takes. Because we can set a varying number of labels required to complete a batch, workers receive $1.50 for short tasks (15 labels) and $10.00 for long tasks (100 labels). We started by releasing shorter batches with more respondents (15 labels per worker) but moved towards longer batches with fewer respondents (100 labels per worker). In our final collection rounds, we paid our best workers $15.00 for 100 labels to show our appreciate for their continued support and work. Amazon charges double (of their typical fee) for tasks that utilize ten or more respondents. So, when we release long batches we release them with 9 respondents to save a bit of money. We provide an example of costs below, by detailing some Amazon Mechanical Turk "receipts" for both a short and long batch.

Table 3.1: Short batch receipt, 60 workers x 15 labels = 900 labels. The total cost with masters qualifications is $130 whereas it is $126 without.

	Value	# of Workers	Cost
Task Reward	$1.50	60	$90.00
Masters Fee (5% of reward)	$0.075	60	$4.50
Amazon Fee (20% of reward)	$0.30	60	$18.00
More than 9 workers Fee (20% of reward)	$0.30	60	$18.00
Total Cost (With Masters)			$130.50
Total Cost (Without Masters)			$126.00

Table 3.2: Long batch receipt, 9 workers x 100 labels = 900 labels. The total cost with masters qualifications is $112 whereas it is $108 without.

	Value	# of Workers	Cost
Task Reward	$10.00	9	$90.00
Masters Fee (5% of reward)	$0.50	9	$4.50
Amazon Fee (20% of reward)	$2.00	9	$18.00
More than 9 workers Fee (20% of reward)	NA	NA	$0.00
Total Cost (With Masters)			$112.50
Total Cost (Without Masters)			$108.00

In both the long and short batches, we receive 900 labeled records. But, in the short batch we receive fewer labeled records per MTurk worker. There are pros and cons to both approaches. Workers are less likely to get labeler fatigue in the shorter batches compared to the longer batches. On the other hand, we had hoped that the label quality would be higher in the longer batches because it would take much more effort

to game the system and would be more challenging to quickly provide poor quality labels. Furthermore, it is easier to pay close attention to worker quality when there are fewer workers. A purely financial motivation to use longer batches is that we don't have to pay an additional 20% to Amazon. After assessing the pros and cons (and taking worker feedback into consideration–see Section 3.3) we decided to focus on releasing 9 person batches where each worker provides 100 labels.

3.1.2 Time Constraints

As shown in Fig. 3.2 we also need to set the time allotted per worker to complete a task. This is the amount of time the worker has between when they accept the task and when they need to submit the task. You want to set a time that is long enough so that workers do not feel stressed / rushed. On the other hand, once a worker accepts a task they get counted towards the total number of respondents. Therefore, we don't want to set the allotted time to be too long or else a worker, who may not end up submitting our task, could occupy a viable slot. For our long tasks (100 labels per worker) we set a time of 2 days, but for short tasks (15 labels per worker) we set a time of 90 minutes. We also have to set a survey expiration date. If tasks are not accepted by this time, they will expire. Additionally, we have to set a time to auto-approve and pay workers. If we don't accept or reject work by this time, it will be automatically accepted. We want to allow ourselves enough time to review the work so that we aren't paying for poor quality work, but we also want to be fair to the workers and make sure they aren't waiting too long to be paid. We decided to go with the recommended amount of time of 3 days.

3.1.3 Worker Requirements

At the bottom of Fig. 3.2 you will see the option to require workers to be "Masters". Masters are high performing workers, as determined by Amazon*. For a typical batch, hiring Masters workers costs an additional $4.50, which we thought was small cost to ensure higher quality labels. But, early analysis of label quality showed that the Masters workers were not higher quality than non-Masters and we still needed to closely monitor and reject many of the works for poor quality work. So, going forward, we removed that qualification from our batch requirements. Amazon MTurk offers numerous other qualifications that we initially did not use for the project.† But, towards the end of the labeling process, we identified our best workers (who seemed to enjoy the tasks) and created batches that only they were invited to complete. We

*"Amazon Mechanical Turk (MTurk) has built technology which analyzes Worker performance, identifies high performing Workers, and monitors their performance over time. Workers who have demonstrated excellence across a wide range of HITs are awarded the Masters Qualification. Masters must continue to pass our statistical monitoring to maintain the MTurk Masters Qualification."[1]

†Selecting workers in a specific location or with specific HIT acceptance rates come at no charge to the requestor. But, to select workers from a specific age group will cost $0.50 per worker. Choosing handedness will cost $0.15 and selecting for smokers will cost $0.30 per worker.

will discuss this further later in the section, but this appeared to be a mutually beneficial decision for both us and the workers.

3.1.4 Example Task

Once we are satisfied with the final version of our batch and have set the proper parameters, we can publish our task. Workers can preview our task, which will look very similar to Fig. 3.3.

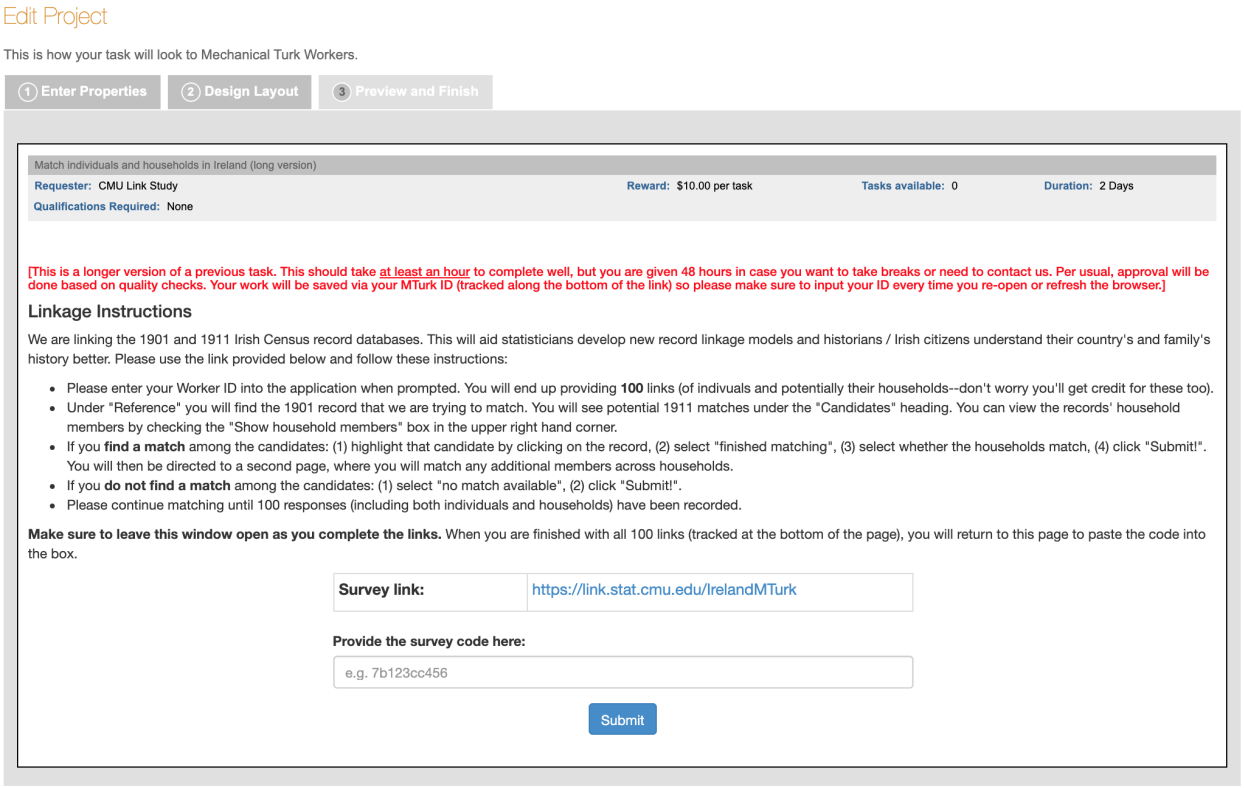


Figure 3.3: A preview of the task we created for MTurk workers.

Workers use the survey link we provided to label records and once they have labeled the designated amount (15 for short batches, and 100 for long batches) they are provided with a survey code in the interface that they can paste into MTurk to submit their work.

3.1.5 Future Tasks

We can simultaneously run multiple batches, but due to computational constraints of our server and of `R Shiny` we tend to keep batch sizes small and only publish one batch at a time. In Fig. 3.4 you can see four different project versions, two short (Ireland4, Ireland6) and two long versions (IrelandLong1, IrelandLong2).

At any point I can publish a batch from any of the projects we've created, or I can repeat the process above to create a new batch.

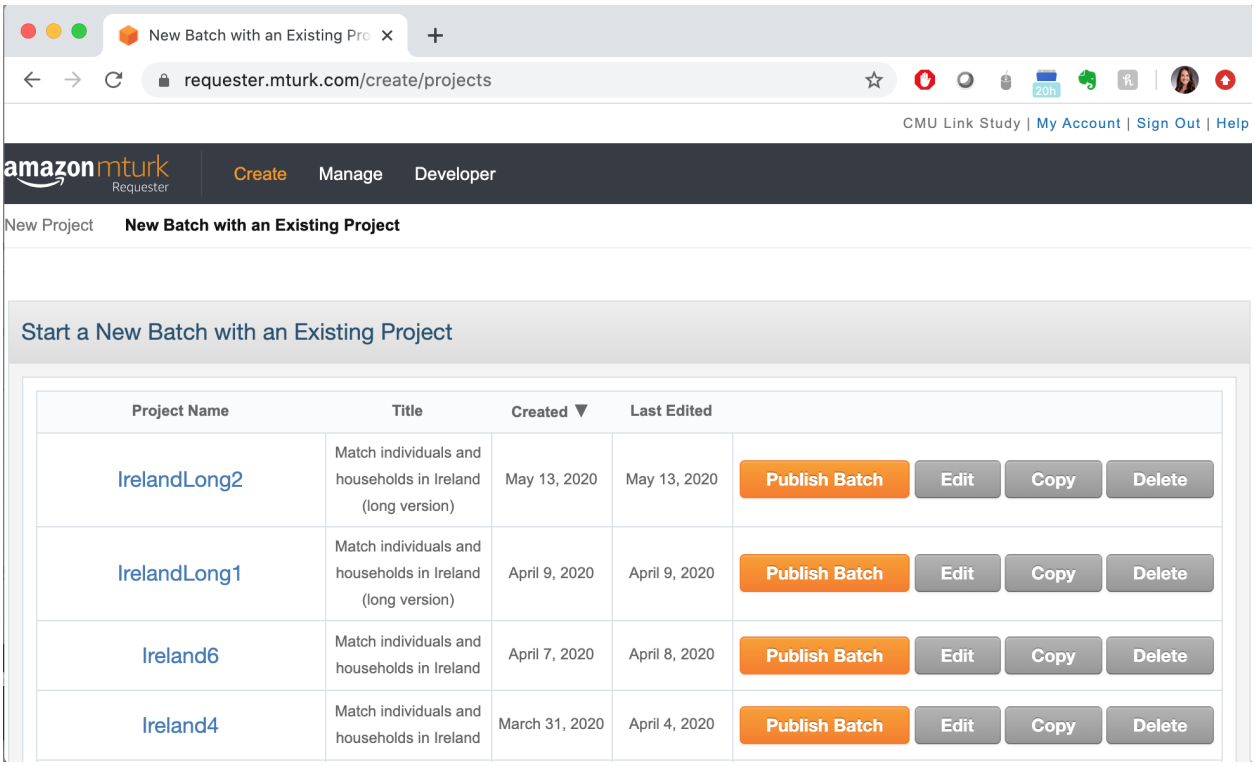


Figure 3.4: Creating an Amazon MTurk batch from an existing batch.

All batches are managed using the "Manage" tab at the top of our MTurk requestor account. In Fig. 3.5 we can see all batches that we have published. We see the batch at the top is in progress and has yet to be completed, but the batch underneath is already finished.

3.2 Work Approval

We have the opportunity to review the submitted work before we decide to pay the workers. Because we believe we are paying the workers fairly and competitively for their time, we only want to accept quality work. Amazon allows us to download a CSV data summary of completed work, which is summarized in Fig. 3.6. We can then upload an edited CSV to quickly approve or reject work in batches. To approve an assignment we put a "x" in the column titled "Approve", and to reject an assignment we put our rejection feedback (required) in the column titled "Reject".

A submission is accepted if the following criteria are met.

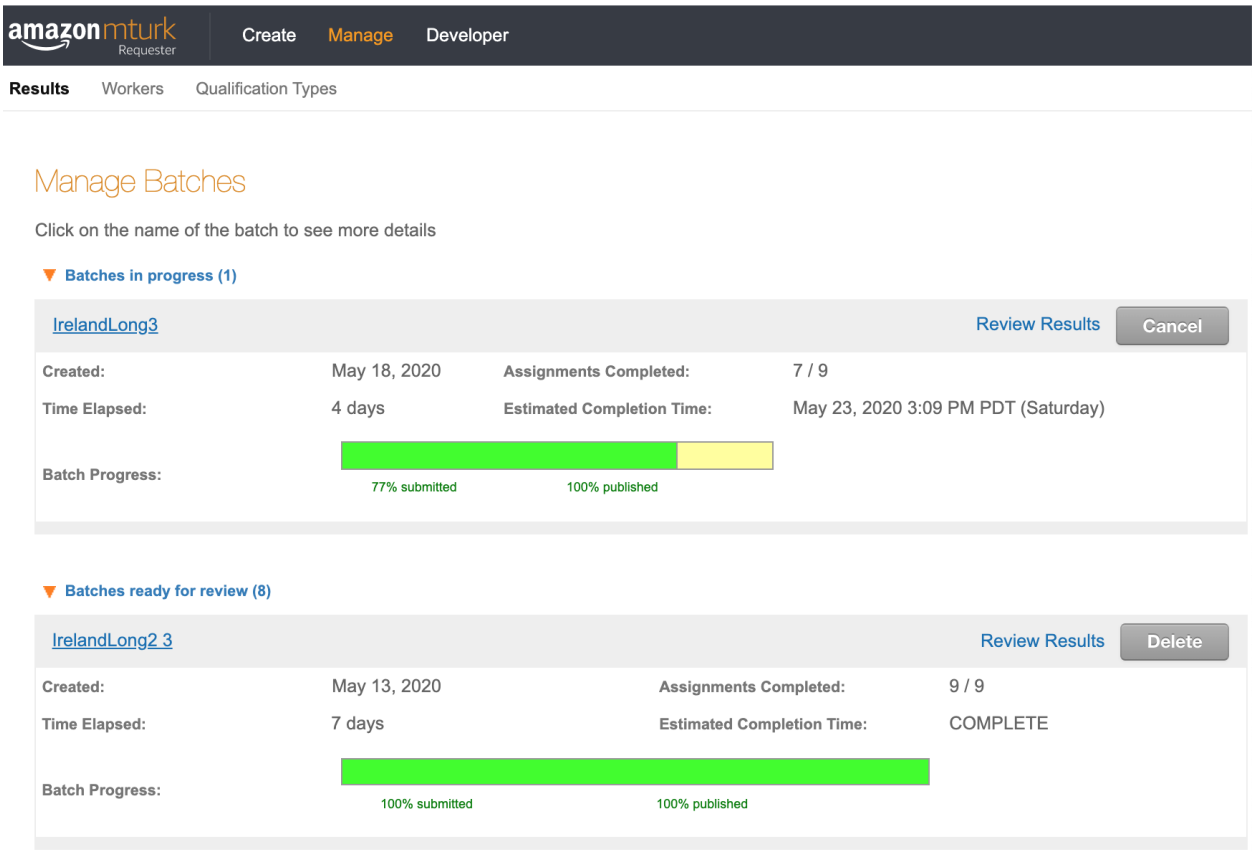


Figure 3.5: Managing existing Amazon MTurk batches.

- The worker labels the correct number of records for a given batch. Note that the correct number of records varies by batch. We have small batches which collect only 15 labels, and larger batches which collect up to 100.
- The survey code the worker provides on MTurk matches the code we give them.
- There is ample time between when the worker accepts and submits the task. This is tracked via AWS and provided in our batch results output. Ample batch times vary by the length of the batch, but for 15 links is 6 minutes.
- There is ample time spent on each individual label. The worker needs to spend at least 10 seconds on at least 75% of their labels.
- The worker clicks on candidates for at least one label.
- The worker selects a candidate (as opposed to selecting "no matches selected") for at least one of their labels.

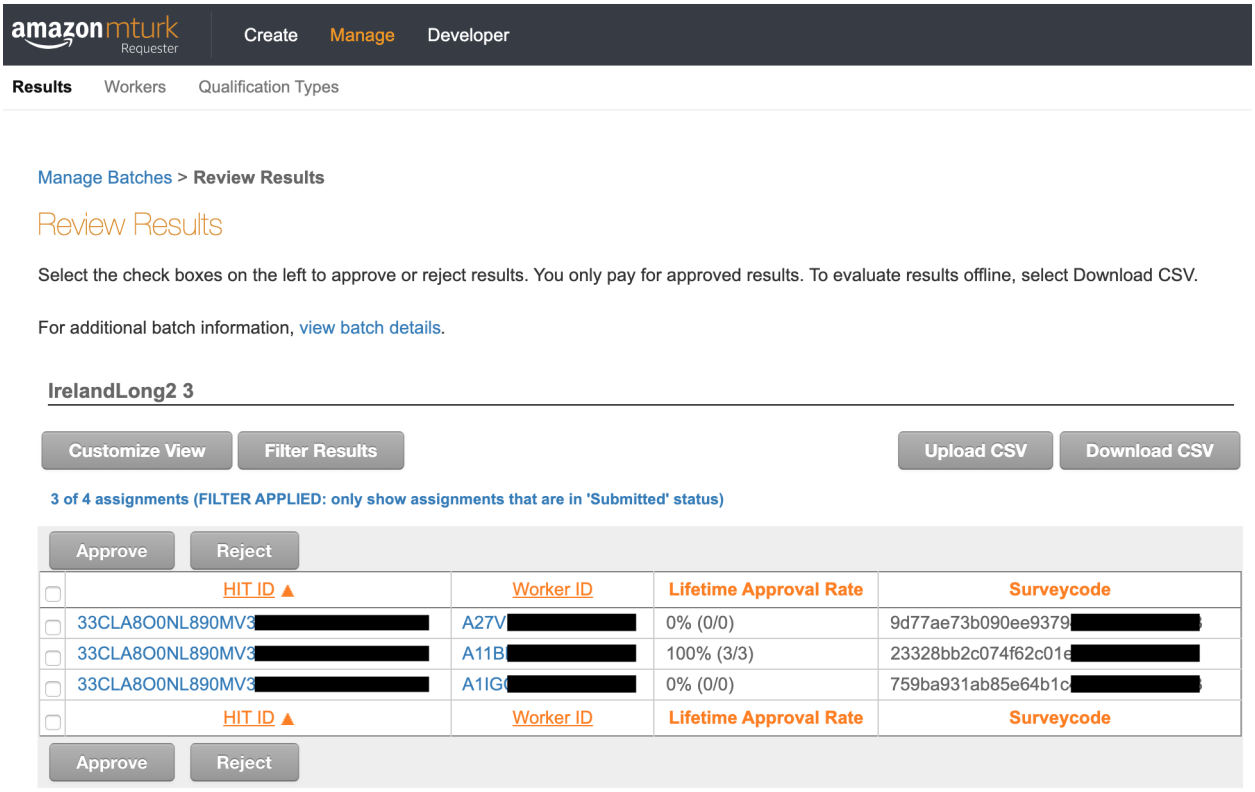


Figure 3.6: A preview of the work batches for review.

Note that these criteria were formed in an iterative approach as we analyzed work quality and interface metadata. When we first moved the application to Amazon Mechanical Turk, we published small batches to better understand the platform and the work quality. We added criteria as we explored and analyzed the MTurk work we received. We found that work quality varied drastically from worker to worker. There were some workers who were hoping that we would simply approve all work and therefore they just made up a survey code and did not click on and open our application. Others figured out the quickest way to move between labels within our application and breezed through the labels as fast as possible to receive a survey code. Those are two examples of poor work quality that would not be accepted, but were achieved in very different ways.

For at least one of their tasks, we rejected work from 70 of the 225 unique workers who submitted tasks with us. The percentage of work rejected decreased to virtually zero as time went on and we adapted our tasks and predominately worked with our best workers. $2,411$ of the $10,647$ submits on Page 1 were by workers who received rejections at some point and 238 of the $2,493$ submits on Page 2 were by rejected workers. We are not surprised that the percentage of rejected work on Page 2 is lower because a common

habit of bad workers is selecting "no candidate selected" for all reference records on Page 1, and never matching records on Page 2.

While work was rejected on the actual Amazon MTurk platform, we do not remove this data from our data base until analyses where we specifically explore label quality (see Chapter 7).

3.2.1 Worker Details

We can also produce reports on the workers who have completed our tasks. Using these reports, we write a script to block bad workers so that they cannot complete future tasks. We can either accept / reject work first within the batch and then proceed to block workers through the "Manage Workers" tab (shown in Fig. 3.7). Or, if we block workers that have current work with us that has yet to be accepted or rejected, we can simultaneously block workers and reject their work.

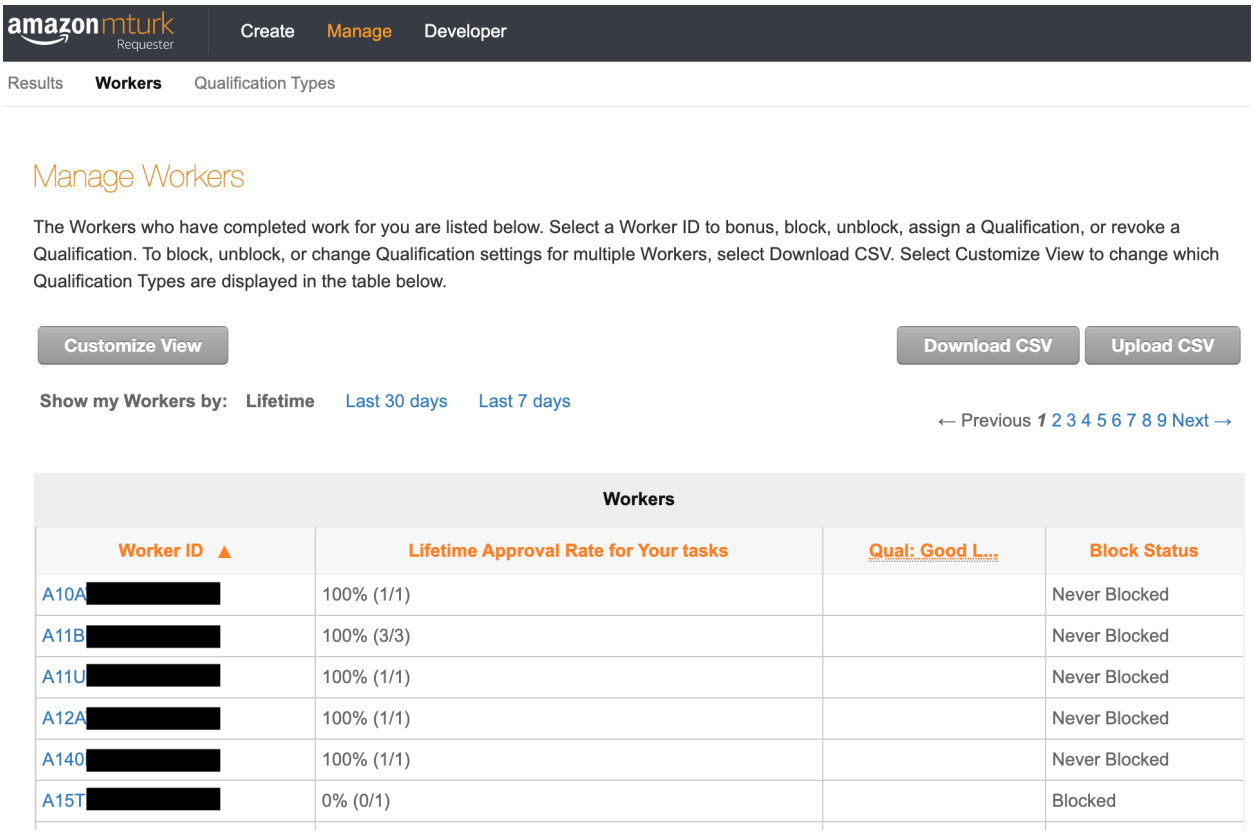


Figure 3.7: Managing MTurk workers by worker ID. Using tools on this page we can block workers or reject work for specific batches. We can do so manually or by uploading a CSV file populated with block / reject information.

Blocking workers ensures that they will not submit any future tasks with us. Let's say we wanted to pay all workers, regardless of work quality. We could still pay workers for their poor quality work and then ensure they don't submit more in future.

3.3 Worker Feedback and Subsequent Changes

Throughout the Amazon MTurk process, we have had to make modifications to the crowdsourcing process and document those here. These changes were made either to better fit MTurk's existing framework, to elicit higher quality labels, or to improve the interface for the MTurk workers.

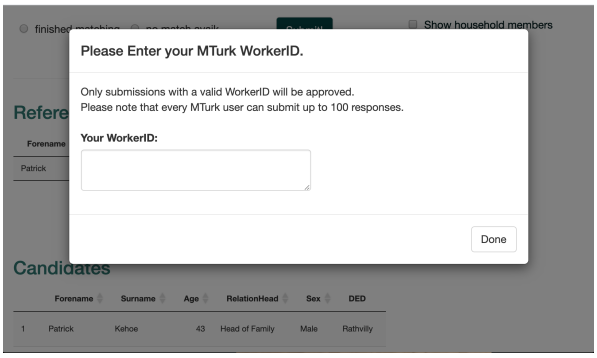


Figure 3.8: Worker ID input. This is how MTurk workers tell the interface who they are so that they can receive credit.

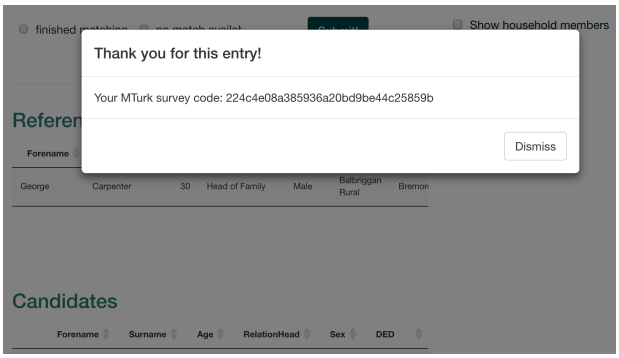


Figure 3.9: Submission code. Workers receive a submission code once they have completed all matches in a batch. We can confirm that they submit a valid / correct code when we review their work.

1	Patrick	Kehoe	43	Head of Family	Male	Rathvilly
2	Patrick	Kehoe	60	Head of Family	Male	Tankardsto
3	Patrick	Kehoe	56	Head of Family	Male	Ticknock
4	Patrick	Kehoe	43	Head of Family	Male	Williamstov

Showing 1 to 4 of 4 entries

Labels this project: 0 Total labels: 57 ID: kfrisoli

Figure 3.10: Interface has been adapted to allow workers to track number of labels they have submitted. This information is highlighted with a red circle.

Moving to an online crowdsourced system, we need to be able to track which worker is labeling what data, so we need the MTurk worker to provide us with their unique MTurk ID. Then, the worker needs to receive and submit a code to receive credit for their work. Both of these changes (Fig. 3.8 and Fig. 3.9) were made to configure our application for the Amazon Mechanical Turk system. These changes also allow workers to submit multiple labels before receiving a code. With traditional MTurk tasks within the pre-formatted templates, you cannot have a worker submit multiple entries as one task. Now that we are using this new system, we needed to show workers how many labels they have already completed. This information is now tracked at the bottom of the interface pages (circled in red at the bottom of Fig. 3.10). It is important to differentiate between a worker's tasks across batches so that we can properly assign credit. These are just some examples of changes we made to the interface to better adapt to the MTurk existing framework.

While working for Amazon MTurk, workers can send the requestor (us) messages. These can be during the middle of a task to ask a clarifying question or to alert us about an application issue. They can also be after we have accepted or rejected their work. The content of these messages varies greatly. We document examples of worker messages below in Table 3.3. The overall sentiment of the positive feedback we receive is that our work is quite different from most of the other tasks. We've also found that those who either have ties to Ireland or have an interest in history / genealogy really enjoy the work. This positive feedback was one of the reasons we decided to release longer batches. We knew we were receiving higher quality labels and that these workers enjoyed the task. Many of them even told us they would enjoy doing even more if they were available.

On the other hand, not all workers enjoyed our task or our standards for work quality. Our negative feedback was mainly from workers who were upset that we rejected their work (Table 3.3, Negative 1). Rejections can negatively affect worker reputations and scores and so we understand why they reach out (and even sometimes ask to re-do the work correctly). Occasionally a worker has a difficult time using our application for a variety of reasons (e.g., server crashes, varying internet speeds). I realized how critical it was to be available to promptly respond to emails while batches are released so that I could personally address any worker challenges or questions. Whenever a user had an issue with the application, I would reply and give them a code to submit so that we can award them for their time, regardless of whether they could finish the task. For example, I responded to Negative Feedback 2 (Table 3.3) with the following message: "I'm so sorry about the issues with the application. Please submit the following code and we will award you for your time: 'INTERFACEerrorCP4720' " but their response was "I've already returned it. You can keep your dollar." Besides realizing how important it is to be attentive to worker emails, I also realized that increasing the allotted completion time (as discussed in Section 3.1.2) can be helpful to allow workers who are having legitimate issues time to debug.

	Positive Feedback
1	Enjoyed the HIT, btw. Nice change of pace from the average survey!
2	I really enjoyed your HIT/task. I recently returned from staying in the U.K. for a few years and found a deep interest in the history and heresy for the common lives of the people, part of my family moved from Scotland in the 1700's and I can see now what they've been through and why we take these large steps as people. I found the same interests when I was looking through the records, seeing how the servants were listed as family in the 'household' and the occupations that entailed change of standing.
3	I thoroughly enjoyed this hit and would be willing to do tons more if available to me! Genealogy is one of my favorite hobbies! Thanks in advance.
4	This is absolutely fascinating, by far the best HIT i ever worked on here! Best regards!
	Negative Feedback
1	Hello sir, I did not attend any qualify test only do 15 matching data.What a reason for reject the hit
2	Your website has numerous coding errors in it. I have wasted an hour trying to help you with your research but the website is unusable. It gives no code though I have made over 15 labels. It continues to bring up the previous record when I match two households. It is slow and time consuming and I am irritated that you have thought so little of my and other users time.
3	Dear Sir, First of all I extremely apologize for the bad work done by me. It's all happened due to some less clarified instructions. I accept it's all mistake done on my side. Again I wish to work on the same HIT with your permission without any issue. Kindly consider to do me the same again.
	Constructive Feedback
1	I wanted to let you know about an issue with your website. I couldn't get any data for household members of potential candidates to load, and couldn't really submit any labels/"no match" entries to the site. I kept getting a delayed error message - something about "ajax 7" and "ajax 9." I really enjoy working on your HITs, and would love to continue on this project, but unfortunately have had to return the HIT due to issues with the website. Please do let me know if you're able to get this resolved - or if there are any steps I can take on my end to get past these errors - I'd love to do more of these HITs. I recently finished my MA in history, so this is the kind of stuff I love to do.
2	When I first started the study, it took really long time for the page to load. However, when it finally did load, I clicked my answer proceeded to submit, but the next page didn't load. I tried refreshing the page, but now I'm getting a proxy error page.

Table 3.3: Questions/Comments/Concerns from MTurk workers who have accepted our tasks

Lastly, messages from workers have been critical in helping us debug the system. Because we are working with new servers (shout-out to our technician Carl Skipper for all of his help) and we often cannot reproduce the same errors locally, worker feedback has been super helpful. Debugging these errors has been a challenging parts of utilizing the MTurk system. We have no idea the workers' computer's age, wifi speeds etc. and therefore have a hard time debugging whether an error is originating on our or their end.

Despite crowdsourcing challenges, Amazon MTurk allowed us to collect data quickly and furthermore allows us to study the impact of data quality on record linkage.

Chapter 4

Data

A large goal of this interface is to generate high quality, record linkage data to aid in the development of linkage methodologies. As mentioned previously, typical record linkage data only contains information about the individual records and binary individual matches (person a matches person b). Data output by the interface, on the other hand, contains additional label information about group structure (i.e., group match, within-group individual matches), label uncertainty, and the human labeling process. A summary of the type of data we collect (which has been described in the previous sections) is shown pictorially in Fig. 2.1's data output section. We have collected data in multiple stages, informally by colleagues as well as formally by utilizing Amazon's Mechanical Turk Platform (Section 3). In this section we summarize the data we have collected, and detail how to take the output data from our interface and prepare/process it for record linkage.

4.1 Data Sources

We first show a sample (both in terms of rows as well as columns) of the original census records. We will specifically focus the labels / data collected for John Clynch (shown in the first row of Table 4.1) throughout this section.

Table 4.1: Original Records

ID	Forename	Surname	Age	Sex	Occupation	Birthplace	TownStreet	DED
1901.68	John	Clynch	31	Male	Gardener/Servant	County Wicklow	Fortgranite	Talbotstown
1911.62	John	Clinch	41	Male	Gardener/Servant	Co Wicklow	Kilmurry	Talbotstown
1911.63	Bridget	Clinch	42	Female	-	Co Carlow	Kilmurry	Talbotstown

4.1.1 Data Collected on Individual Matches

As detailed in Section 2.1 we collect information / data on both Page 1 and Page 2 of the interface. We will use the phrases "Page 1" and "Page 2" to distinguish between labels / data collected about individuals directly on the first page versus as part of household matches on the second page. These data are stored separately and previewed in Tables 4.2 and 4.4, respectively.

One row of data in Tables 4.2 is the information that is collected after a labeler clicks "Submit" on Page 1. We collect the ID of the 1901 reference (Reference) record that we are labeling, the ID of the candidate that the labeler chose (Candidate) as well as the ID of the labeler (LabelerID). We asked the labeler whether the households of the individuals matched (House Match) and we collected the sequence of their clicks on various candidates (Click Sequence). We store within which round the label was collected (Round). The labels originate from one of six rounds which we call: CMU_round1, CMU_round2, MTurk_initial, MTurk, MTurkLong, and MTurkLongSubset. These rounds are labeled in chronological order as we first utilized students, staff, and faculty at CMU to label the records and then moved to the Amazon MTurk platform. As detailed in Chapter 3 we adapted our label collection on Mechanical Turk to collect labels using fewer labelers in longer tasks. In the final round (MTurkLongSubset) we utilized our best workers for a long task on only subsets of records that had already been previously labeled (to ensure we collected uncertainty for those records). We additionally store the time that the label was collected. We can map this data back to the original census records if we wanted to know, for example, to which household the reference and candidate belongs. We also have information about which candidates were shown to the labeler when labeling a given reference record. The process was refined between cycles, and any information that is not available (likely because it was not collected at the time) is denoted with a dash "-".

Table 4.2: Page 1 Interface Data for 1901 Record `1901.68`

Reference	Candidate	LabelerID	House Match	Click Sequence	Round	Timestamp	...
1901.68	1911.62	a05cb	-	-	CMU_round1	-	...
1901.68	1911.62	ebe29	1	4	CMU_round2	2019-01-21 14:38:14	...
1901.68	no_matches_selected	d31ab	0	2_1_4_2_4	MTurk	2020-03-27 12:58:39	...

In Table 4.2 we see all labels for the reference record `1901.68` of 1901 John Clynch (one row = one submit). John was labeled by three separate labelers on Page 1 in three separate labeling rounds. The first two labelers determined that candidate record `1911.62` was the correct match while the third labeler did not think that there was a match among the candidates shown. We were not collecting click data when the first instance was labeled, but we see that labeler 2 did not click / view other candidates while the third labeler clicked around before selecting "No Match". This small example is consistent with what we see across the full data collection; the amount of time / number of clicks varies greatly by labeler.

Because the information shown in Table 4.2 is only about the candidate record that was chosen, we need to expand this data to include all of the 1911 candidates who were *not* selected. We show the expanded labels from Page 1 in Table 4.3. We now need to add a column for whether the individual pairs were matched as the same individual ("Match"). In cases where the labeler said "no_matches_selected" we need to store a non-match (zero) for all candidates that were shown to the labeler, but ultimately not selected. There are some fields (i.e. LabelerID, Round) that are consistent across all candidates for a given label (even those that were not selected). Some fields (e.g., House Match) only relate to the candidate that was linked to the reference; in the case of House Match we need to assign a household non-match (zero) to non-selected candidates. Now that we have done an initial preprocessing of labels from Page 1, we need to explore the data that are collected on the second page of the interface.

Table 4.3: Full Page 1 Interface Data for 1901 Record `1901.68`

Reference	Candidate	Match	LabelerID	House Match	Click Sequence	Round	...
1901.68	1911.96	0	a05cb	-	-	CMU_round1	...
1901.68	1911.1080	0		-			...
1901.68	1911.59	0		-			...
1901.68	1911.62	1		-			...
1901.68	1911.96	0	ebe29	0	4	CMU_round2	...
1901.68	1911.1080	0		0			...
1901.68	1911.59	0		0			...
1901.68	1911.62	1		1			...
1901.68	1911.96	0	d31ab	0	2_1_4_2_4	MTurk	...
1901.68	1911.1080	0		0			...
1901.68	1911.59	0		0			...
1901.68	1911.62	0		0			...

In Table 4.4 we see the data collected on Page 2 of the interface. The record `1901.68` was additionally labeled three more times on Page 2 of the interface. This means that other people from John's households were linked on Page 1 and therefore John was shown as a potential within-house match on Page 2. We can see that this occurred when three of his 1901 housemates (`1901.70`, `1901.71`, `1901.69`) were labeled on Page 1 (meaning they appear as the original, Page 1 References in Table 4.4). John was then shown on the left of Page 2 and compared to all of the 1911 household members on the right. John (`1901.68`) was matched with `1911.62` by labeler `65454`, `1911.63` by labeler `de566`, and `1911.62` by labeler `60f7d`. A zero is recorded for all of the "Match" column for 1911 individuals within the household that John was not matched to. There are many other matches and non-matches across the two households that are not shown here. We only show the pairs related to John Clynch (`1901.68`).

As with data collected on Page 1, on Page 2 we also collect information about who is labeling the records, what round the label originated in, and the timestamps of the collection. These variables are shown in the last columns of Table 4.4.

Table 4.4: Page 2 Interface Data for 1901 Record `1901.68`

Reference	Candidate	Match	Page 1 Reference	Page 1 Candidate	LabelerID	Round	Timestamp	...
1901.68	1911.62	1	1901.70	1911.196	65454	CMU_round2	2019-01-21 11:48:17	...
1901.68	1911.63	0						...
1901.68	1911.64	0						...
1901.68	1911.66	0						...
1901.68	1911.67	0						...
1901.68	1911.62	0	1901.71	1911.66	de566	MTurkLong	2020-04-12 11:09:17	...
1901.68	1911.63	1						...
1901.68	1911.64	0						...
1901.68	1911.65	0						...
1901.68	1911.67	0						...
1901.68	1911.62	1	1901.69	1911.63	60f7d	MTurkLongSubset	2020-08-30 05:02:58	...
1901.68	1911.64	0						...
1901.68	1911.65	0						...
1901.68	1911.66	0						...
1901.68	1911.67	0						...

4.1.2 Interface Data Consolidation

Once we have collected data on both Page 1 and Page 2 of the interface we need to consolidate the labels and more importantly the matches. We present a sample of the consolidated labels for reference record `1901.68` in Table 4.5. We consolidate the data separately for labels collected on Page 1 (at the individual level) and on Page 2 (at the within-household level). The column "Match" represents the number of times the pair was labeled as a match and the "Total" represents the total number of times the pair was seen by a labeler (and therefore had the ability to be matched). As a reminder, the reason why we have multiple labels at both the individual and household level is that we allow multiple labelers to label the same record. Notice that there are three sources for this data and the source tells us where the label(s) originated from. To this matrix we can calculate and add the similarities between the fields of the reference and candidate records as well. We will detail this in Chapter 5.

There are numerous ways that we can utilize the raw numbers of matches and total labels. The first way is to simply take the proportion of times that a candidate was chosen by dividing the matches by the totals. We can do this for both Page 1 and Page 2 separately, as well as together. We can then use this overall proportion and determine a binary yes/no match by letting anything over 0.5 be a match and anything below be a non-match. We see that we have two pairs that both get classified as a match when we do this. If we wanted to enforce that there is only one match for each reference, we could assign a 1 to the pair that received the most matches and a 0 to all other pairs. In this case `1911.62` received four matches and `1911.63` only received one. Despite this difference, the Page 2 and Total Match Proportions make the two candidates

Table 4.5: Individual Pairwise Label Consolidation. Un-processed match numbers are shown in the Match and Total columns. We identify where the label was created using the Source column.

Reference	Candidate	Page 1 Match	Page 1 Total	Page 2 Match	Page 2 Total	Source	Field Similarities
1901.68	1911.62	2	3	2	3	Page 1 & 2	...
1901.68	1911.96	0	3	-	-	Page 1	...
1901.68	1911.1080	0	3	-	-	Page 1	...
1901.68	1911.59	0	3	-	-	Page 1	...
1901.68	1911.64	-	-	0	3	Page 2	...
1901.68	1911.65	-	-	0	2	Page 2	...
1901.68	1911.66	-	-	0	2	Page 2	...
1901.68	1911.67	-	-	0	3	Page 2	...
1901.68	1911.63	-	-	1	2	Page 2	...

seem similarly likely. We need to think about situations like this when we proceed with future analysis of the data (e.g., modeling matches). Note that we can use the raw counts to calculate other combinations of these values (e.g., weight matches from Page 1 higher than Page 2). We propose rather naive approaches to label consolidation because we are interested in understanding the record linkage labeling process in its rawest form, but there is work suggesting the use of statistical models to assign weights to labelers pre-consolidation [45] [18].

Table 4.6: Individual Pairwise Match Consolidation. We include the match proportions (match / total) for Page 1 and Page 2 separately. We then consolidate these in the column "Total Match Proportion". We can discretize whether the total match proportion is greater than 0.5. We can also enforce that only one candidate receives a 1 per reference record.

Reference	Candidate	Page 1 Match Proportion	Page 2 Match Proportion	Total Match Proportion	Total Match?	Max Candidate?
1901.68	1911.62	0.67	0.67	0.67	1	1
1901.68	1911.96	0.00		0.00	0	0
1901.68	1911.1080	0.00		0.00	0	0
1901.68	1911.59	0.00		0.00	0	0
1901.68	1911.64		0.00	0.00	0	0
1901.68	1911.65		0.00	0.00	0	0
1901.68	1911.66		0.00	0.00	0	0
1901.68	1911.67		0.00	0.00	0	0
1901.68	1911.63		0.50	0.50	1	0

Besides information about the number of matches and the number of labels, we also have auxiliary information about the reference / candidate pairs. We detail some of these other fields in Table 4.7. For each of the pairs in Table 4.5 we see the IDs of the labelers from both pages as well as the round in which

the label was collected. We also have information about how often the pair's houses matched and similarity scores for the pair's household.

Table 4.7: Individual Pairwise Labels - Meta Data. Here we preview information that we have for each reference and candidate pair. We know where each pair was labeled and we know whether or not their households were also linked. Not shown, but we have information on time spent and number of clicks for each pair as well.

Reference	Candidate	Page 1 Labelers	Page 2 Labelers	Page 1 Rounds	Page 2 Rounds	House Matches and Field Similarities
1901.68	1911.62	a05cb, ebe29, d31ab	60f7d, 65454, de566	CMU round1, CMU round2, MTurk	MTurkLongSubset, CMU round2, MTurkLong	...
1901.68	1911.96	a05cb, ebe29, d31ab	d4cd0	CMU round1, CMU round2, MTurk	-	...
1901.68	1911.1080	a05cb, ebe29, d31ab	d4cd0	CMU round1, CMU round2, MTurk	-	...
1901.68	1911.59	a05cb, ebe29, d31ab	d4cd0	CMU round1, CMU round2, MTurk	-	...
1901.68	1911.64	d4cd0	60f7d, 65454, de566	-	MTurkLongSubset, CMU round2, MTurkLong	...
1901.68	1911.65	d4cd0	60f7d, de566	-	MTurkLongSubset, MTurkLong	...
1901.68	1911.66	d4cd0	60f7d, 65454	-	MTurkLongSubset, CMU round2	...
1901.68	1911.67	d4cd0	60f7d, 65454, de566	-	MTurkLongSubset, CMU round2, MTurkLong	...
1901.68	1911.63	d4cd0	65454, de566	-	CMU round2, MTurkLong	...
1901.68	1911.103	d4cd0	d4cd0	-	-	...
⋮	⋮	⋮	⋮	⋮	⋮	...

This meta deta about the pairwise matches can be used to understand label uncertainty as well as analyze specific subsets of our data (e.g., CMU versus MTurk labelers).

4.1.3 Matching Households

After the first round of labeling at CMU, we started collecting whether or not the labeler thought that the two households matched. Because we allowed labelers to view and utilize household data to make selections

for the individuals, we soon realized that labelers were spending a lot of time studying the households. Therefore, because they were already spending quality time examining the household, we decided that we should capture whether the households matched and any extra individual matches within those households. We discuss this in more detail when we introduced Page 2 in Section 2.3. Additionally, familial-based record linkage data seldom contains information about whether the households match so this information adds depth / uniqueness to our data. CITE Using this additional information provided by our labelers, we can calculate whether or not two households match as well as field similarities about the two households.

Table 4.8: Labels that make Household Comparisons.This is the subset of the individual pairwise data for anyone from h(1901.68) and h(1911.62). Labels from individual pairs are consolidated to understand whether households should be considered matches. Households can be compared via any of the individuals that reside in that household.

Reference	Candidate	Page 1 Match	1901 House	1911 House	House Match	LabelerID	Round
1901.70	1911.65	1	h(1901.70)	h(1911.65)	1	65454	CMU_round2
1901.69	1911.63	1	h(1901.69)	h(1911.63)	1	4365a	CMU_round2
1901.68	1911.62	1	h(1901.68)	h(1911.62)	1	ebe29	CMU_round2
1901.68	1911.62	0	h(1901.68)	h(1911.62)	0	d31ab	MTurk
1901.70	1911.65	0	h(1901.70)	h(1911.65)	0	e5cd9	MTurk
1901.71	1911.66	1	h(1901.71)	h(1911.66)	1	de566	MTurkLong
1901.69	1911.63	1	h(1901.69)	h(1911.63)	1	60f7d	MTurkLongSubset

We collect labels for household matches on Page 1 of the interface. When a labeler submit a (potential) matching candidate, they also determine if the candidate's household matches the reference's household. We pull all of the Page 1 labels for the two households of interest (John / John's Matches): **Household 1:** $h(1901.68) = h(1901.69) = h(1901.70) = h(1901.71) = h(1901.72)$ and **Household 2:** $h(1911.62) = h(1911.63) = h(1911.64) = h(1911.65) = h(1911.66) = h(1911.67)$ and display these labels in Table 4.8. As a reminder, h is a function that pulls the household ID for each of the individual records; because `1911.62` and `1911.63` are in the same household, $h(1911.62) = h(1911.63)$.

Now that we have all information about household matches for these two households (meaning all Page 1 labels for anyone from the two households), we can start to define what it means for a house to match. One way we can consider two houses to match is by summing the number of times that the household matched, given that the candidate was chosen (definition 1). Using definition 1, a house would only receive a zero (non-match) if a candidate within the household was chosen but the household was not matched. We can see in Table 4.8 that the first three and last two rows fall into this category in that they received a one (1) for the Page 1 Match. Out of these five rows, all of them received a one (1) for the House Match meaning that this household pair had 5/5 matches when using definition 1. This definition, however, doesn't capture the cases where a labeler explored the two households as part of a Page 1 match but decided not to link the individuals (and therefore couldn't possibly link the households).

In definition 2 all of the households of the candidates that were *not* selected will also receive a zero when determining House Match. The denominator (total labels for definition 2) becomes the count of the times two households were shown to a labeler on Page 1 instead of just the times that they were shown and an individual was matched. The numerators in both definitions are equal. This process of determining what counts as a non-match is subjective and there are pros and cons to both definitions. The main concern with Definition 2 is that there could be a situation where the household actually matched even though the candidate shown did not match the reference shown.

Table 4.9: Household Comparisons. We take the unprocessed, subset data about houses h(1901.68) and h(1911.62) in Table 4.8 and consolidate that information here.

1901 House	1911 House	House Match	House Total (Def 1)	House Total (Def 2)	Def 1 Proportion	Def 2 Proportion	Field Similarities
$\begin{pmatrix} \text{h(1901.68)} \\ =\text{h(1901.69)} \\ =\text{h(1901.70)} \\ =\text{h(1901.71)} \\ =\text{h(1901.72)} \end{pmatrix}$	$\begin{pmatrix} \text{h(1911.62)} \\ =\text{h(1911.63)} \\ =\text{h(1911.64)} \\ =\text{h(1911.65)} \\ =\text{h(1911.66)} \\ =\text{h(1911.67)} \end{pmatrix}$	5	5	7	1	0.71	...

In Table 4.9 we show what a row of our household comparison (one row = one pair of households) looks like for the household's of individual records `1901.68` and `1911.62`. We see the sum of the labels using both definitions as well as the consolidated proportion of matches across the two definitions. "Def 1 Proportion" is defined as House Match / House Total (Def 1) and "Def 2 Proportion" is defined as House Match / House Total (Def 2). We can also calculate the field similarities for variables at the household level (see Section 5.3 for details). The household of `1901.68` is also compared to many other households but those data are not shown here.

Tables 4.5, and 4.6 preview how labels about individual pairs originating on Page 1 and Page 2 of our interface are processed and consolidated. Table 4.9 shows this process for pairs of households.

4.2 Blocking

The pairs of labels that we saw previewed in Table 4.5 came directly from our labeling interface. We only include labels (matching or non-matching) for records that were shown to a labeler on either page 1 or page 2. More specifically, the pair was either

- labeled as a match on Page 1,
- labeled as a non-match (by not being selected) on Page 1,
- labeled as a match on Page 2 (by being linked across households), or

- labeled as a non-match on Page 2 (by not being linked across households).

We understand that in many record linkage settings one is not collecting their own labels and wants to model the data using existing labels. In these situations, when a researcher is attempting to model pre-labeled record linkage data, it is very common to "block" the records ahead of time. As mentioned in Section 1.1.1, blocking is a common tool used in record linkage to increase computational traction and address the large class imbalance. There's not always a need to compare 1901 "Mary Murphy" to every single person from 1911. Instead we may only want to compare her to the likely 1911 matches. Records are only compared if they match a set of criteria. Any pairs that don't meet this criteria are never compared (and therefore could never be predicted as matches). There are many ways to build blocks using unsupervised or supervised methods and these blocks can be independent from each other or built through passes. A record linkage paper that links historical records from Canada block using a first name code and the first letter of the last name [8]. If a researcher were to take our pre-labeled data (which we will make public) and decide to block the records for a pairwise analysis, they might take a similar approach to [8] due to the similarities in application.

Note that we have already imposed strict blocking by limiting the number of candidates we show our labelers. Showing a human labeler hundreds of records to parse through would be inefficient and likely produce low quality labels. Given our initial strict blocking, we are additionally interested in how record linkage models perform when there is less strict blocking applied to our data. Therefore, we need to add in blocked pairs for the reference records we've already labeled. For a given reference record, additional candidates are selected if they:

Table 4.10: Blocking Criteria. In additional blocked pairs, we block based on the soundex of first name, the first letter of the last name, and geographic region.

Blocking Criteria
Agree on the first two letters of the Soundex representation of first name,
Agree on the first letter of last name, and
Live within a bordering county.

We utilize this blocking scheme to mimic other historical record linkage blocking schemes. We needed to add the additional location constraint to our blocks to reduce the block size further. There is an extremely large name similarity across Ireland in this time period and the first two conditions did not constrain the size enough. These additional blocked pairs will all have zero "Matches" meaning that they were never labeled as a match. We feel comfortable assigning these blocking pairs to be non-matches because the reference record had the ability to or was linked in our interface. We do not add additional blocks to pairs that were labeled as non-matches on Page 2 but were never labeled on Page 1.

We add additional blocked, non-matching pairs to the John Clynch example that we've used throughout this chapter (shown in Table 4.11). We add these because John was matched on both Page 1 and Page 2, meaning that we feel comfortable assigning a zero to additional blocking pairs.

Table 4.11: Individual Pairwise Matches - Adding Blocking. This is a representation of the rows of data that are added via blocking.

Reference	Candidate	Page 1 Match	Page 1 Total	Block Match	Block Total	Source
1901.68	1911.62	2	3	-	-	Page 1 & 2
1901.68	**1911.103**	-	-	0	1	Block 1
1901.68	**1911.117**	-	-	0	1	Block 1
1901.68	**1911.132**	-	-	0	1	Block 1
⋮	⋮	⋮	⋮	⋮	⋮	

As shown in Table 4.12 the records `1911.103`, `1911.117`, and `1911.132` all fit the blocking criteria and are somewhat similar records to John (`1901.68`). Note that these records didn't match strongly enough to be considered a candidate in our interface. From examining the original records, we feel confident that none of these individuals match better to John than the John that was selected most often by our labelers.

Table 4.12: Blocking Records. The original data for a few of the records that are added to the comparison space for `1901.68`.

ID	Forename	Surname	Age	Sex	Occupation	Birthplace	TownStreet	DED
1911.103	James	Candy	54	Male	Farm Servant	Co Carlow	Bromville	Ballintemple
1911.117	John	Clarke	20	Male	Farm Servant	Dublin City	Kilgraney	Ballintemple
1911.132	James	Coe	60	Male	General Servant	Co Carlow	Craans	Ballintemple

We add blocks to help understand the effect of block strictness (on future modeling), given that there is no established "correct" or "standard" blocking scheme in practice. One of the benefits of looser (less strict) blocking is the potential to find matches among the additionally added data. While we assigned a zero as the true match value for all blocked pairs, it would be interesting to see whether any models uncover mistakes or blocked pairs that should have been examined and matched earlier in our process. In future sections we make the distinction between models / analyses that were used with pr without the additional pairs from "blocking".

4.3 Data Summary

We show the number of labels we collected via crowdsourcing in Table 4.13. CMU Round 1 was collected in March 2018, and Round 2 was collected in January 2019. We collected our initial MTurk labels in December of 2019. The rounds "MTurk", "MTurkLong", and "MTurkLongSubset" were collected in March, April, and June of 2020. We tabulate the number of labels by the round in which the label was collected. Given the design of our interface, we explore the number of labels separately colleced on Page 1 and Page 2. Looking at the row for Round "MTurkLongSubset" our labelers submitted labels on Page 1 $2,251$ times. Of those submits, $1,200$ found a matching candidate and $1,004$ additionally found a matching household. On Page 2, labelers in "MTurkLongSubset" provided labels for 3616 reference records or reference/candidate pairs within the $1,004$ household matches. As we show in Section 2.6.1 and can see here, there are label quantity gains by labeling on Page 2 of the interface given that the average number of labels per household pair is greater than 1. In the last two rows under the dashed line, we total these values and also provide the number of unique labels. There were $13,058$ submits on Page 1 equating to $8,531$ unique reference record. The remainder were duplicate labels. On Page 2 we received $2,731$ submits and of those 2031 were unique household pairs. The remainder household pairs were labeled multiple times. On Page 2 we received labels for $8,850$ individual pairs of which 6034 were for unique 1901 records.

Table 4.13: Number of labels collected via crowdsourcing. We see the breakdown of labels collected in each of the rounds of labeling. The rounds of labeling are ordered chronologically. We first started labeling at CMU and then moved to MTurk. The first two columns represent a submit on Page 1 of the interface. When someone submits on Page 1 they give a label (or say no match selected) for one reference record. The second column represents the submits on Page 1 where a match was found among the candidates (the labeler did *not* say no match selected). On Page 2 the labeler can link multiple pairs of people across the two households. Therefore for each household submit, we can receive multiple matching labels. The fourth column represents the total number of 1901 records that were matched on Page 2. At the bottom under the dashed line, we see the total and unique number of labels. Because we have multiple labelers label a given reference record, it is important to distinguish between total labels and unique labels.

	Page 1 Submits	Page 1 Match Found	Page 2 Submits	Page 2 Match Found
CMU_round1	2737	1928	0	0
CMU_round2	786	561	474	1823
MTurk_initial	767	181	96	325
MTurk	2983	709	341	1029
MTurkLong	3534	1251	816	2057
MTurkLongSubset	2251	1200	1004	3616
Total	13058	5838	2731	8850
Total (Unique)	8531	3666	2031	6034

In Table 4.14 we explore how the data we collect changes by round. We first explore the percentage of times that a labeler said that there was no match found. One of the biggest concerns when we started utilizing the Amazon MTurk platform was how often labelers said that there was no match among the provided candidates (potentially introducing false negatives into the data). Looking once again at the last round, "MTurkLongSubset", we find that $1 - (1200/2251) = 0.467$ or 46.7% of submits said "no match selected". In this round, the labelers said the household matched about 45% of the time ($1004/2251 = 0.446$). When they labeled on Page 2, they provided labels for on average 3.6 reference records per household. There are clear differences between CMU and MTurk labeler behavior, which we will explore further in the modeling sections.

	No Match Found %	House Match %	Average Individual Labels Per House
CMU_round1	29.60		
CMU_round2	28.60	60.30	3.85
MTurk_initial	76.40	12.50	3.38
MTurk	76.20	11.40	3.02
MTurkLong	64.60	23.10	2.52
MTurkLongSubset	46.70	44.60	3.60

Table 4.14: The column "No Match Found%" is the percentage of submits on Page 1 in which the labeler said that there was no match among the presented candidate records. The column "House Match %" is the percentage of labelers on Page 1 who said that the households matched (and were therefore sent to Page 2). The column "Average Individual Labels Per House" is the average number of matching pairs we received for each submit on Page 2.

The summary tables 4.13 and 4.14 do not include the pairs of non-matching individuals that we receive when a labeler utilizes our interface. To describe the full number of pairs received we look to Table 4.15. There were over 90,000 individual pairwise labels collected on Page 1 and almost 5,000,000 on page 2. Additionally, we received labels for 58,000 pairs of households.

Table 4.15: Pairs of labels collected via crowdsourcing.

	Page 1 Individual Pairs	Page 2 Individual Pairs	Household Pairs	Block Individual Pairs
Match (1)	5830	8467	2218	0
Non-Match (0)	84588	4952533	56264	12390803
Total	90438	4961000	58482	12390803

In Figure 4.1 we examine the distribution of blocks added to each reference record. We find that for reference records with very common names, up to almost $8,000$ rows of data could be added. When modeling we take steps to examine how results change with only subsets of our added blocks, but this will be discussed in more detail later.

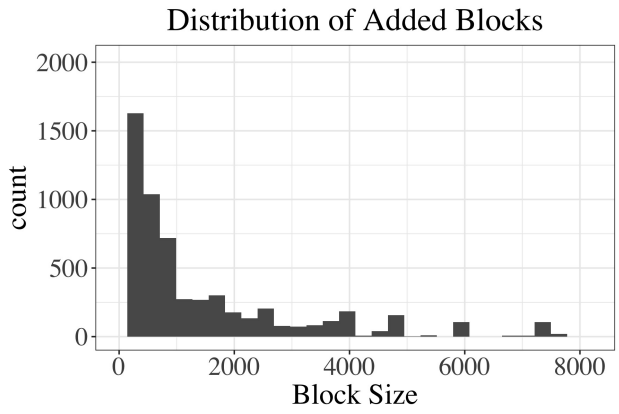


Figure 4.1: Distribution of block size for each reference record. Most records have less than 1000 added blocks but some reference records have up to 8000.

[[Kayla says: Add a map of Ireland with a dot for each label]]

4.3.1 Data Going Forward

Our initial work and data collection was focused on Ticknock, County Carlow, Ireland (and surrounding areas) which is a small rural town in the southeastern part of the country. As we move to later phases of label collection, we expand to other geographic regions including Counties Carlow, Wicklow, Kilkenny, Dublin, Kildare, and Meath. In addition to collecting larger geographic regions, we also collected matching information at the household level about households and individuals within households. For following analyses, we will focus on two data sets that were collected with labelers at both CMU and Amazon MTurk. The first data set represents matches/non-matches that we receive directly from the labeling interface. The second contains additional non-matching pairs pulled using a typical blocking approach. The data includes the larger geographic region we mentioned above.

Chapter 5

Comparing Data

In this Chapter, we detail similarity metrics for record linkage. In Section 5.1 we introduce notation that we will use throughout the remainder of the thesis. In Section 5.2 we introduce comparison metrics for comparing pairs of individuals. In Section 5.3 we introduce comparison metrics for comparing pairs of groups of individuals. In Section 5.4 we mathematically compare the group-wise comparison metrics and briefly conclude in Section 5.5.

5.1 Notation

Individuals & Individual Pairs

Given a set of census records from year t we use the notation $X^t_{i,k}$ to denote the kth feature of the ith individual within year t. Within each year, there are $i = 1...n_t$ individuals and $k = 1...K_t$ features. In this thesis, we are working with two sets of census records, one from 1901 and one from 1911. We refer to these data sets as $\boldsymbol{X}^{1901}$ and $\boldsymbol{X}^{1911}$ respectively. There are n_{1901} records with K_{1901} features in the first data set and n_{1911} records with K_{1911} features in the second. Therefore, $\boldsymbol{X}^{1901}$ is an $(n_{1901} \times K_{1901})$ matrix and $\boldsymbol{X}^{1911}$ is a $(n_{1911} \times K_{1911})$ matrix. Let $\boldsymbol{X}^{1901}_i$ and $\boldsymbol{X}^{1911}_j$ index records in $\boldsymbol{X}^{1901}$ and $\boldsymbol{X}^{1911}$, respectively.

If we wanted to compare all individuals within 1901 to all individuals within 1911, we would need to make $n_{1901} \cdot n_{1911}$ pairwise comparisons. These pairs can be classified either as a match (i.e., they truly refer to the same entity) or a non-match (i.e., they do not refer to the same entity). We use $\boldsymbol{X}^{1901}_i \sim \boldsymbol{X}^{1911}_j$ to denote that records i and j are a true match and belong to the same entity. We use M to refer to the set of all pairs of records that truly match and U to refer to the set of pairs that are true non-matches, such that $\boldsymbol{M} = \{(\boldsymbol{X}^{1901}_i, \boldsymbol{X}^{1911}_j) \in \boldsymbol{X}^{1901} \times \boldsymbol{X}^{1911} : \boldsymbol{X}^{1901}_i \sim \boldsymbol{X}^{1911}_j\}$ and $\boldsymbol{U} = \{(\boldsymbol{X}^{1901}_i, \boldsymbol{X}^{1911}_j) \in \boldsymbol{X}^{1901} \times \boldsymbol{X}^{1911} : \boldsymbol{X}^{1901}_i \nsim \boldsymbol{X}^{1911}_j\}$. We use lower case $\boldsymbol{m}$ and $\boldsymbol{u}$ to refer to matches and non-matches within a subset of record

pairs. When comparing a subset of 1901 records ($\boldsymbol{A}$) to a subset of 1911 records ($\boldsymbol{B}$), we define the matches to be: $\boldsymbol{m} = \{(\boldsymbol{a}_i, \boldsymbol{b}_j) \in \boldsymbol{A} \times \boldsymbol{B} : \boldsymbol{a}_i \sim \boldsymbol{b}_j\}$.

Households & Household Pairs

For all records, we were provided with a household identifier that indicates at which household the individual record was recorded at. The function $h^t(i) : \{1, \ldots, n_t\} \mapsto \{1, \ldots, n_{h^t}\}$ maps individual indices to household indices (e.g., $i \mapsto h^{1901}(i)$). For example, $h^{1901}(i)$ represents the household ID for person i from the year 1901. We use $\boldsymbol{X}^t(h^t(i))$ to refer to the subset of all records that are in the household of person i from time t: $\boldsymbol{X}^t(h^t(i)) \subset \boldsymbol{X}^t = \{\boldsymbol{X}^t_{l\cdot} : h^t(l) = h^t(i)\}$.

Let $h^{1901}(i)$ and $h^{1911}(j)$ index households in $\boldsymbol{X}^{1901}$ and $\boldsymbol{X}^{1911}$, respectively. Furthermore, we may be interested in the set of one specific field within a household. We use $\mathbb{X}^t_{\cdot k}(h^t(i)) = \{X^t_{lk} : h^t(l) = h^t(i)\}$ to denote the set of individual values for field k within house $h^t(i)$ (of year t). We provide concrete examples of how our notation is used below.

Notation Examples

For the remainder of this chapter, we will use a running example (Table 5.1) to illustrate various comparison metrics. In this example we have the Murphy household from 1901 containing a father, mother, and two daughters. The 1911 household only contains the two parents. This is a situation where the daughters moved out of the house between the ten year census recording gap. Therefore the daughters do not appear in the 1911 census, even though the households are the same.

Example 1901 Household

Individual Index (i)	House Index (h_a)	Forename	Surname	Gender	Age	$\cdots$
32	6	Patrick	Murphy	Male	45	$\cdots$
33	6	Ellen	Murphy	Female	42	$\cdots$
34	6	Eliza	Murphy	Female	15	$\cdots$
35	6	Mary	Murphy	Female	12	$\cdots$

Example 1911 Household

Individual Index (j)	House Index (h_b)	Forename	Surname	Gender	Age	$\cdots$
9596	2401	Patrick	Murphy	Male	56	$\cdots$
9597	2401	Ellen	Murphy	Female	43	$\cdots$

Table 5.1: Examples of the Murphy Households from 1901 and 1911. We believe that Patrick and Ellen are the same individuals in 1901 as 1911, but that Eliza and Mary moved between the two census' and therefore do not appear in 1911.

We are often interested in comparing individuals across data sets. Using the example households in Table 5.1, we may be interested in comparing Ellen Murphy's 1901 record ($\boldsymbol{X}^{1901}_{33,\cdot}$) to her 1911 record ($\boldsymbol{X}^{1911}_{9597,\cdot}$). To compare these records as a whole, we may want to compare them field by field. For example, if we wanted to compare the Surnames of both records, we could compare $X^{1901}_{33,4}$ to $X^{1911}_{9597,4}$. $X^{1901}_{33,4}$ is "Murphy", which is the 4th feature (Surname) of the 33rd record in the 1901 census. $X^{1911}_{9597,4}$ is also "Murphy", which is the 4th feature (Surname) of the 9597th record in the 1911 census. Besides comparing individuals, sometimes we want to compare entire households. In a similar fashion, we may want to compare all first names of one household to another. For example, we could compare $\mathbb{X}^{1901}_{\cdot 3}(h_6)$ (which we can also write as $\mathbb{X}^{1901}_{\cdot \text{Surname}}(h_6)$) to $\mathbb{X}^{1911}_{\cdot 3}(h_{2401})$. This would allow us to compare the set {Patrick, Ellen, Eliza, Mary} to the set {Patrick, Ellen}. We will provide further examples of metrics to compare individuals and households in the sections below.

5.2 Comparing Individuals

5.2.1 Exact Match Similarity

One metric for comparing two strings is to determine whether or not they are an exact match, which we define formally below.

$$sim_{\text{exact-match}}(s_1, s_2) = \begin{cases} 0 & \text{if } s_1 \neq s_2 \\ 1 & \text{if } s_1 = s_2 \end{cases} \tag{5.1}$$

For example, we can compare the forenames of record pairs from Table 5.1. We first compare $X^{1901}_{32,3}$ to $X^{1911}_{9596,3}$. Both entries for the 3rd field (Surname) are the string "Patrick", which are an exact match and therefore are calculated to have a score of "1". Next we compare $X^{1901}_{32,3}$ (Patrick) to $X^{1911}_{9597,3}$ (Ellen) which are not the same string, so that pair receives a score of "0" for the binary, exact match comparison.

$$sim_{\text{exact-match}}(\text{"Patrick"}, \text{"Patrick"}) = 1$$
$$sim_{\text{exact-match}}(\text{"Patrick"}, \text{"Ellen"}) = 0$$

Exact match comparisons can be made for both string / text and numeric fields, which make this comparison metric easy to apply in many settings. The binary nature of the metric is also easy to work with, although exact matching can be seen as too stringent in many record linkage settings because much of our data is recorded with typographical errors or inconsistencies that exact match metrics will not consider.

5.2.2 Numeric Valued Differences

A common approach to comparing numeric values is to take their difference. Age is a common example for which utilizing a difference metric makes sense, but we need to be cognizant of applications with a longitudinal time component. For example, our two census data bases were recorded 10 years apart and therefore we expect individuals to age 10 years in that time frame. If we are using the difference metric in a continuous way, we are able to use the raw difference score, defined below. We subtract the 1901 value from the 1911 value.

$$sim_{\text{difference}}\left(X_{i,k}^{1901}, X_{j,k}^{1911}\right) = X_{j,k}^{1911} - X_{i,k}^{1901} \tag{5.2}$$

Below, we subtract Patrick's 1901 age from his 1911 age from our running example (Table 5.1).

$$sim_{\text{difference}}\left(45, 56\right) = 11$$

Absolute Value Differences

If we know we expect a certain value difference (e.g., ten years between a decennial Census), we may want to create our own difference metrics. For example, one metric we found useful was the number of years away from the expected value of ten. The intuition is that we may want to treat a 9 year difference between ages the same as an 11 year difference. Using the "years from ten" difference value below, both of those differences will be the same.

$$sim_{\text{years-from-ten}}\left(X_{i,k}^{1901}, X_{j,k}^{1911}\right) = \left|10 - \left(X_{j,k}^{1911} - X_{i,k}^{1901}\right)\right| \tag{5.3}$$

Below, we apply the "years from ten" difference to Patrick's 1901 and 1911 ages. If the two ages were exactly ten years apart, the difference would produce zero.

$$sim_{\text{years-from-ten}}\left(45, 56\right) = \left|10 - \left(56 - 45\right)\right| = \left|10 - 11\right| = 1$$

Binned Differences

As we mentioned previously, sometimes we want binary comparisons because of their simplicity. However, historical evidence has shown (and our research has confirmed) that census ages are wildly inaccurate and therefore a binary exact match on age (even after accounting for the 10 year difference) may not be practical. Therefore, for age in particular, we can 1) adjust temporally, 2) bin our data, and 3) apply an exact match to utilize a less strict exact match.

Therefore, for age in particular, we could subtract any known temporal diferences (e.g., 10 years), bin those values, and then see if the bins are an exact match across the pair we want to pair. This is one way to apply a looser exact match for age. We show this below:

$$sim_{\text{binned-difference}}\left(X_{i,k}^{1901},\, X_{j,k}^{1911}\right) = sim_{\text{exact-match}}\left(\text{bin}(X_{i,k}^{1901}),\, \text{bin}(X_{j,k}^{1911} - 10)\right). \tag{5.4}$$

Binning can be done in any way, but we chose to bin our data into 5 year intervals from 0 to 110. Once again we apply this difference to the Patrick example from Table 5.1, and show the results here:

$$sim_{\text{binned-difference}}\left(45, 56\right) = sim_{\text{exact-match}}\left(\text{bin}(45),\, \text{bin}(56 - 10)\right) = sim_{\text{exact-match}}((45, 50], (45, 50]) = 1$$

.

5.2.3 Jaro-Winkler Similarity

As formerly noted in Section 1.1.1, a common metric for assessing the similarity of two strings is an edit distance called the Jaro-Winkler (JW) [64]. It is simliar to the Jaro similarity[34], but places a higher emphasis on letters within the strings matching at the beginning of the string. Both of these similarity scores are bounded between 0 and 1 with 1 indicating the highest similarity and 0 the lowest. We might use Jaro-Winkler to assess how similar "suzanne" is to "susan" or how similar "catholic" is to "church of england". The Jaro-Winkler allows us to assign a continuous or fuzzy match metric to two strings (which the Exact Match similarity did not). We define the Jaro-Winkler edit distance below formally, for the comparison of two strings s_1 and s_2, where $s_1 = X_{i,k}^{t_1}$, and $s_2 = X_{j,k}^{t_2}$. We use this notation to notate that we are interested in comparing field k (e.g., forename) between the ith individual at time t_1 (e.g., 1901) and the jth individual at time t_2 (e.g., 1911).

$$sim_{\text{jaro}}(s_1, s_2) = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{c}{|s_1|} + \frac{c}{|s_2|} + \frac{c-t}{c}\right) & \text{otherwise} \end{cases} \tag{5.5}$$

Where:

- $|s_1|$ is the number of characters in the string s_1
- c is the number of matching characters
- t is the number of transpositions

Using the Jaro Similiarity (Equation 5.5), we define the Jaro-Winkler Similarity (Equation 5.6) below. Once again, letting $s_1 = X_{i,k}^{t_1}$ and $s_2 = X_{j,k}^{t_2}$,

$$sim_{\text{jaro-winkler}}(s_1,\ s_2) = sim_{\text{jaro}}(s_1,\ s_2) + lp(1 - sim_{\text{jaro}}(s_1,\ s_2)) \tag{5.6}$$

Where:

- l is the number of characters that match at the beginning of both strings, with a maximum of 4 characters
- p is a constant scaling factor, often $p = 0.1$

More details on the Jaro and Jaro-Winkler similarity and the definitions of matching characters and transpositions can be found on the "Jaro–Winkler distance" Wikipedia page* or the Statistical Odds & Ends Blog†.

5.2.4 Consolidating Field Similarity Into Individual Pairwise Comparisons and Matches

Now that we know how to compare fields across two records (e.g., Exact Match, Difference, Jaro-Winkler), we want to consolidate the field similarities and try to determine whether the two records actually belong to the same entity.

There are many ways to determine if two individuals match. To highlight a few ways we could match the individuals, we use the Patrick Murphy example shown in Table 5.1 as well as below (Table 5.2). We decide on a similarity metric for each field (in practice one could calculate multiple metrics per field) and show the similarity scores below the raw data (Table 5.3).

*https://en.wikipedia.org/wiki/Jaro-Winkler_distance

†https://statisticaloddsandends.wordpress.com/2019/09/11/what-is-jaro-jaro-winkler-similarity/

Patrick Murphy Raw Census Data

Year	Individual Index (i)	House Index (h_a)	Forename	Surname	Gender	Age	Occupation	Birthplace
1901	32	6	Patrick	Murphy	Male	45	Farmer	Co Carlow
1911	9596	2401	Patrick	Murphy	Male	56	Agriculture	Carlow

Table 5.2: Patrick Murphy original census records.

Patrick Murphy Similarity Comparisons

Forename: J-W	Surname: J-W	Gender: Exact	Age: Binned Exact	Occupation: J-W	Birthplace: J-W
1	1	1	1	0.5	0.8

Table 5.3: Patrick Murphy similarity comparisons.

Matches can be determined by heuristics or rules. For example we could say if all fields are their max similarity, classify as a match. In the Patrick Murphy example, the Jaro-Winkler (J-W), Exact, and Binned Exact similarities are all bounded by a maximum similarity value of 1. Since two of the fields are not the max value we would classify the pair as a non-match. A rule that would classify the pair as a match might be: if name, gender and age all exactly match and the other fields have a similarity of at least a 0.5 level, classify as a match. Multiple sets of rules can be used together, joined by either "and" or "or" statements. These sets can be developed by exploring the data (either in an unsupervised or supervised fashion) and / or by working with experts in the application area.

As another example, various pairwise field similarity metrics could be averaged or summarized. If all similarities were bounded between 0 and 1, we could simply take the average. In our example, we would calculate an overall pairwise similarity of $\frac{1+1+1+1+0.5+0.8}{6} = 0.883$. We could then decide that all pairs that have a similarity above some threshold, say 0.7 be declared a match. Both of the methods described above are unsupervised in that no ground truth match status data is used to determine whether or not the pair is a true match. Other, more statistical, unsupervised linkage approaches will be discussed in Chapter D.

Statistical models could also be used to classify individuals as matches in a pairwise fashion. For example, we could predict (0/1) whether an individual pair is a match from the variables we generate about their pairwise fields using the true known match status of other pairs. In a later Chapter (see 6), we go into much greater details about statistical models for record linkage.

As mentioned in our notation section (Section 5.1), we use $\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}$ to refer to two individuals who truly are the same entity (in our case they are the same person recorded in both 1901 and 1911). However, we do not always know whether or not two individuals are the same person. In our application, the ground truth linkage structure is not known. We do collect labels via our human label collection, but that is for only a subset of the data. Therefore, we will also predict whether or not two individuals are a match. We will use a "hat" to indicate that a matching link is estimated: $\boldsymbol{X}_i^{1901} \hat{\sim} \boldsymbol{X}_j^{1911}$. The full set of estimated matches is

indicated by $\hat{\boldsymbol{M}} = \{(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911}) \in \boldsymbol{X}^{1901} \times \boldsymbol{X}^{1911} : \boldsymbol{X}_i^{1901} \hat{\sim} \boldsymbol{X}_j^{1911}\}$. Similarly for subsets of individual matches, we use $\hat{\boldsymbol{m}} = \{(\boldsymbol{a}_i, \boldsymbol{b}_j) \in \boldsymbol{A} \times \boldsymbol{B} : \boldsymbol{a}_i \hat{\sim} \boldsymbol{b}_j\}$ to indicate the estimated matches.

5.3 Group Similarity

In order to expand similarity from individuals to groups of individuals, we need to establish and utilize metrics for comparing groups. These metrics and methods are often different from those used to compare individuals. In this subsection, we outline existing group similarity approaches.

5.3.1 Jaccard Index

A common mathematical approach to comparing sets, or groups of items, is the Jaccard index (Jaccard similarity coefficient) [33]. This index is bounded between 0 and 1 where 1 indicates the maximum similarity and 0 indicates the least. We define the Jaccard index between two, unordered sets, $\mathbb{A}$ and $\mathbb{B}$ in Equation 5.7.

$$\text{Let } \mathbb{A} = \{a_1, a_2, ..., a_{n_a}\}$$
$$\text{Let } \mathbb{B} = \{b_1, b_2, ..., b_{n_b}\}$$
$$\text{Jaccard}(\mathbb{A}, \mathbb{B}) = \frac{|\mathbb{A} \cap \mathbb{B}|}{|\mathbb{A} \cup \mathbb{B}|} \tag{5.7}$$

For example, we could compare the set of first names across the two households in Table 5.1. Using the same notation introduced earlier, we can compare $\mathbb{A} = \mathbb{X}^{1901}_{\cdot\text{Forename}}(h_6) = \{\text{Patrick, Ellen, Eliza, Mary}\}$ to $\mathbb{B} = \mathbb{X}^{1911}_{\cdot\text{Forename}}(h_{2401}) = \{\text{Patrick, Ellen}\}$ and calculate a Jaccard index of $\frac{2}{4}$ because "Patrick" and "Ellen" appear in both sets (intersection size of two) and there are four unique elements across both sets.

$$\mathbb{A} = \{\text{ Patrick, Ellen, Eliza, Mary}\}$$
$$\mathbb{B} = \{\text{Patrick, Ellen}\}$$
$$\text{Jaccard}(\mathbb{A}, \mathbb{B}) = \frac{|\mathbb{A} \cap \mathbb{B}|}{|\mathbb{A} \cup \mathbb{B}|} = \frac{|\{\text{Patrick, Ellen }\}|}{|\{\text{Patrick, Ellen, Eliza, Mary}\}|} = \frac{2}{4} = 0.5$$

5.3.2 Ruzicka / Weighted Jaccard Index

An extension of the Jaccard Index, otherwise known as an "adjusted" or "weighted" Jaccard index has been re-introduced numerous times across the literature for various applications[50][12][Deza]. This "weighted" version is known as the Ruzicka similarity, the Jaccardized Czekanowski Index, or the Generalized Jaccard format[47]. If $n_a = n_b$ and all $a_i, b_i \geq 0$, we can write the Jaccard index (Equation 5.7) using the Ruzicka similarity (Equation 5.8) below. Note that the numeric valued sets $\mathbb{A}$ and $\mathbb{B}$ are paired/ordered (unlike the sets in Equation 5.7) such that a_1 is compared to b_1 and a_{n_a} is compared to b_{n_b}. The specific order of elements within $\mathbb{A}$ and $\mathbb{B}$ does not matter, as long as $\mathbb{A}$ can directly be compared to $\mathbb{B}$, element by element.

$$\text{Jaccard}_{\text{ Ruzicka}}(\mathbb{A}, \mathbb{B}) = \frac{\sum_i \min\left(a_i, b_i\right)}{\sum_i \max\left(a_i, b_i\right)} \tag{5.8}$$

One disadvantage of the Ruzicka similarity is that (because of the additional constraints on our sets) we would not be able to calculate the similarity of the previous example ($\mathbb{A}$ = {Patrick, Ellen, Eliza, Mary} to $\mathbb{B}$ = {Patrick, Ellen}) because the sets are different lengths, the values are non-numeric, and it is unclear how to pairwise compare the elements across the two sets. As a human, we may be able to guess that the "Patrick"s and "Ellen"s should be compared and that "Eliza" and "Mary" do not have a pair in 1911, but a computer or automated system would not know which elements to compare. As a reminder, the Ruzicka similarity metric can only be used (as is) if we were to compare two sets that each had multiple, positive numeric features (in an order such that the sets can be compared element-wise). However, if we were to calculate the counts of each word across $\mathbb{A}$ and $\mathbb{B}$, we could create ordered, positive numeric sets and satisfy the constraints of the Ruzicka. We call these tabulations $\mathbb{A}'$ and $\mathbb{B}'$, which are the counts of the following elements: {Eliza, Ellen, Mary, Patrick}. For this example we arbitrarily chose to order the text alphabetically, but we could have ordered them in a different way. It only matters that we compare the count of "Eliza" from $\mathbb{A}'$ to the count of "Eliza" from $\mathbb{B}'$ and not that we compare the "Eliza" counts first and the "Mary" counts third. The fact that we are only comparing the 1901 "Eliza" to the 1911 "Eliza" (as opposed to another name) is another benefit of the Ruzicka. If we know the inherent ordering within the sets, we can focus on the comparisons of elements that are "paired" or should be directly compared. Now that we have $\mathbb{A}' = \{1, 1, 1, 1\}$ and $\mathbb{B}' = \{0, 1, 0, 1\}$ representing the counts of the unique words in both sets, we can apply the Ruzicka similarity to assess how similar $\mathbb{A}$ is to $\mathbb{B}$. Note that because these elements are counts, if "Eliza" happened to appear twice in the first set we would see a "2" instead of a "1" as the first element. Both of the hypothetical "Eliza"s would get to contribute to the Ruzicka similairty, whereas the unadjusted Jaccard does not change with the addition of repeated elements.

$$\mathbb{A}' = \{1, 1, 1, 1\}$$

$$\mathbb{B}' = \{0, 1, 0, 1\}$$

$$\text{Jaccard}_{\text{ Ruzicka}}\ (\mathbb{A}', \mathbb{B}') = \frac{\sum\limits_{i} \min\left(a'_i, b'_i\right)}{\sum\limits_{i} \max\left(a'_i, b'_i\right)} = \frac{0+1+0+1}{1+1+1+1} = \frac{2}{4} = 0.5$$

We notice that the Ruzicka Jaccard of $\mathbb{A}'$ and $\mathbb{B}'$ produce the same score as the unadjusted Jaccard of $\mathbb{A}$ and $\mathbb{B}$, but this is not always the case. We will discuss their differences further in Section 5.3.3.

5.3.3 Adjusting the Jaccard Index for Record Linkage

We noticed in the above subsections that there are benefits to using the Ruzicka similarity, but that it cannot be used in all settings (as is). Therefore, we formally rewrite our sets such that we can always use either the Jaccard or the Ruzicka similarity. The first step is to refer to our sets as multisets, given that there can be repeated elements. Multisets have been used for hundreds of years within the mathematics literature, but was formally coined in the 1970s [35]. A multiset is a set in which elements can be repeated. The support of a multiset is the set of unique elements of the muiltiset. The support is also known as the underlying set of the multiset. In a multiset, we care about the number of times each element of the support appears in the multiset. The multiplicity of an element in the support is the number of times the set element appears in the multiset. The cardinality of the multiset is the total number of elements within the set. We define these formally below.

$$\mathbb{A} = \text{a multiset} \tag{5.9}$$

$$Multiplicity(a) = d_{\mathbb{A}}(a) = \sum_{x \in \mathbb{A}} \mathbb{1}(x = a) \tag{5.10}$$

$$Multiplicity(\mathbb{A}) = \{d_{\mathbb{A}}(a) \mid a \in Supp(\mathbb{A})\} \tag{5.11}$$

$$Support(\mathbb{A}) = Supp(\mathbb{A}) = \{a \in U \mid d_{\mathbb{A}}(a) > 0\} \tag{5.12}$$

$$Cardinality(\mathbb{A}) = |\mathbb{A}| = \sum_{a \in Supp(\mathbb{A})} d_{\mathbb{A}}(a) \tag{5.13}$$

Now that we have the language to refer to these sets, we can rewrite the Jaccard and the Ruzicka similarities below, using this terminology. If we are working with multisets, but want to capture the original

Jaccard (Equation 5.7), we can write the Jaccard as follows. Note that $d_{Supp(\mathbb{A})}(x)$ is either going to be 1 (if $x \in \mathbb{A}$) or 0 (if $x \notin \mathbb{A}$). We further note that $x \in Supp(\mathbb{A}) \cup Supp(\mathbb{B})$ is the same as $x \in \mathbb{A} \cup \mathbb{B}$, meaning all x in the union of all unique values across $\mathbb{A}$ and $\mathbb{B}$. Because both unions are the same and the summations below could be written in either way, we choose the simpler one. In the original Jaccard equation we only look to the support of the set because we do not want to incorporate information about repeated elements.

$$\begin{aligned} \text{Jaccard}(\mathbb{A}, \mathbb{B}) &= \frac{|A \cap B|}{|A \cup B|} \\ &= \frac{\sum\limits_{x \in \mathbb{A} \cup \mathbb{B}} \min\left(d_{Supp(\mathbb{A})}(x), d_{Supp(\mathbb{B})}(x)\right)}{\sum\limits_{x \in \mathbb{A} \cup \mathbb{B}} \max\left(d_{Supp(\mathbb{A})}(x), d_{Supp(\mathbb{B})}(x)\right)} \end{aligned} \tag{5.14}$$

Alternatively, when we compare two sets and *do* want to incorporate information about the repeated elements, we look to the full multiset (we calculate the multiplicity of the full mutiset as opposed to the multiplicity of the support). The equations are equivalent except in the Ruzicka Jaccard we replace the multiplicity of the support of the multiset ($d_{Supp(\mathbb{A})}(x)$) with the multiplicity of the multiset ($d_{\mathbb{A}}(x)$).

$$\begin{aligned} \text{Jaccard }_{\text{Ruzicka}}(\mathbb{A}, \mathbb{B}) &= \frac{\sum\limits_{i} \min\left(a_i, b_i\right)}{\sum\limits_{i} \min\left(a_i, b_i\right)} \\ &= \frac{\sum\limits_{x \in \mathbb{A} \cup \mathbb{B}} \min\left(d_{\mathbb{A}}(x), d_{\mathbb{B}}(x)\right)}{\sum\limits_{x \in \mathbb{A} \cup \mathbb{B}} \max\left(d_{\mathbb{A}}(x), d_{\mathbb{B}}(x)\right)} \end{aligned} \tag{5.15}$$

Example

In the section above (5.3.3) we introduce a format for referring to multisets, such that we can easily compute both the Jaccard and the Ruzicka Adjusted Jaccard. We saw that when assessing the similarity of first names from Table 5.1, the Jaccard and the Ruzicka produced identical results. However, for variables with repeated elements (i.e., $\mathbb{A} \neq Support(\mathbb{A})$) the results are not always the same. To demonstrate this idea, we compare the `Gender` of the households in our recurring example (Table 5.1).

$$\mathbb{A} = \{Male, Female, Female, Female\}$$
$$Supp(\mathbb{A}) = \{Male, Female\}$$
$$Multiplicity(\mathbb{A}) = (1, 3)$$
$$Multiplicity(Supp(\mathbb{A})) = (1, 1)$$
$$Cardinality(\mathbb{A}) = 4$$

$$\mathbb{B} = \{Male, Female\}$$
$$Support(\mathbb{B}) = \{Male, Female\}$$
$$Multiplicity(\mathbb{B}) = (1, 1)$$
$$Multiplicity(Supp(\mathbb{B})) = (1, 1)$$
$$Cardinality(\mathbb{B}) = 2$$

$$\mathbb{A} \cup \mathbb{B} = \{Male, Female\}$$

Now that we have determined the support and multiplicity of our multisets, we can calculate the Jaccard and the Ruzicka Jaccard.

$$\text{Jaccard}(\mathbb{A}, \mathbb{B}) = \frac{\sum_{x \in \mathbb{A} \cup \mathbb{B}} \min\left(d_{Supp(\mathbb{A})}(x), d_{Supp(\mathbb{B})}(x)\right)}{\sum_{x \in \mathbb{A} \cup \mathbb{B}} \max\left(d_{Supp(\mathbb{A})}(x), d_{Supp(\mathbb{B})}(x)\right)} = \frac{1+1}{1+1} = \frac{2}{2} = 1.0$$

$$\text{Jaccard}_{\text{ Ruzicka}}(\mathbb{A}, \mathbb{B}) = \frac{\sum_{x \in \mathbb{A} \cup \mathbb{B}} \min\left(d_{\mathbb{A}}(x), d_{\mathbb{B}}(x)\right)}{\sum_{x \in \mathbb{A} \cup \mathbb{B}} \max\left(d_{\mathbb{A}}(x), d_{\mathbb{B}}(x)\right)} = \frac{1+1}{1+3} = \frac{2}{4} = 0.50$$

The `Gender` example above exemplifies how having repeated elements within a set can inadvertently produce a higher Jaccard index than we may expect. The two multisets $\mathbb{A}$ and $\mathbb{B}$ are not identical, yet they produce the maximum similarity score using the Jaccard index. In variables with many repeated elements (e.g., gender, surname) the Jaccard index is often 1, because it does not consider the repeated elements and only the unique ones. In the above example, the Jaccard index does not account for the two additional Female individuals in 1901 that do not appear in 1911. Instead, we calculate an index of 1, which we know is

the maximum similarity value. While the Jaccard index may work well for some fields, for others that often have repeated values within household members, the Jaccard index (5.7) will dilute or ignore the information we have about repeated elements within households.

5.3.4 Group Linkage Measure

So far we have covered the comparison of individual fields and how to consolidate those into the comparison of individual records (Section 5.2). We have also explored the similarity of sets of individual field values at the household level (Section 5.3). However, we sometimes want to asses how similar two households are based on the similarity of individuals within those households. In 2007, On, Koudas, Lee, and Srivastava introduced the Group Linkage Measure as a way to link groups when individual similarities and matches within groups are known [43]. Group linkage gives one similarity value per household pair whereas the Jaccard similarities give one similarity value per field per household pair. One application example they use is the linkage of authors (group variable) across various databases where the author has multiple citations (each citation is an individual record) within each database. Each citation includes features such as the title and co-author(s). Their goal is to link as many authors as possible, using the group information available from the citations. Authors are linked if there are enough matching citations between them and the strength of the similarity between citations is high enough. To better illustrate group linkage, we define it below and follow with an example.

The group linkage measure is the similarity of Group A ($\boldsymbol{A}$) to Group B ($\boldsymbol{B}$). Each group is made up of the individual records $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$, respectively, such that $\boldsymbol{a}_i$ is the ith individual / record from group $\boldsymbol{A}$. $\hat{\boldsymbol{m}}$ is the pairs of individuals / records across groups $\boldsymbol{A}$ and $\boldsymbol{B}$ that have been estimated to be true matches. The "hat" on top of the similarity symbol indicates that these matches are estimated and are not ground truth. Similarity of record pairs across the two groups can be made in a variety of ways including, but not limited to, heuristics, edit distances, TF-IDF cosine similarity, and model outputs (e.g., predicted probability of matching)[49][36]. The numerator of Equation 5.16 is the sum of these individual pairwise similarity scores, but only for pairs that are considered to match (typically by an arbitrary cutoff for the similarity scores). The numerator is the total number of estimated unique individuals. It is calculated by summing the number of individuals in group A and group B and subtracting those individuals who were double counted across the groups (i.e. those who are actually the same person).

$$\boldsymbol{A} = \{\boldsymbol{a}_{1,\cdot}, \boldsymbol{a}_2, ..., \boldsymbol{a}_{n_{h^{1901}(i)}}\}$$
$$\boldsymbol{B} = \{\boldsymbol{b}_{1,\cdot}, \boldsymbol{b}_2, \boldsymbol{b}_3, ..., \boldsymbol{b}_{n_{h^{1911}(j)}}\}$$
$$\hat{\boldsymbol{m}} = \{(\boldsymbol{a}_i, \boldsymbol{b}_j) \in \boldsymbol{A} \times \boldsymbol{B} \,:\, \boldsymbol{a}_i \hat{\sim} \boldsymbol{b}_j\}$$

$$sim_{\text{group-linkage}}(\boldsymbol{A}, \boldsymbol{B}) = \frac{\sum\limits_{(\boldsymbol{a}_i, \boldsymbol{b}_j) \in \hat{\boldsymbol{m}}} sim(\boldsymbol{a}_i, \boldsymbol{b}_j)}{|\boldsymbol{A}| + |\boldsymbol{B}| - |\hat{\boldsymbol{m}}|} \tag{5.16}$$

As an example, let's compare the households in Table 5.1. We use $\boldsymbol{A}$ and $\boldsymbol{B}$ to represent the household records, with $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$ denoting the individual records within the two households, respectively. We can set $\boldsymbol{A} = \mathbb{X}^{1901}(h_6) =$ The 1901 Murphy Household and $\boldsymbol{B} = \mathbb{X}^{1911}(h_{2401}) =$ The 1911 Murphy Household. As a reminder of notation from Section 5.1, $\boldsymbol{A} = \boldsymbol{X}^t(h^t(i)) = \{\boldsymbol{X}^t_{l\cdot} : h^t(l) = h^t(i)\}$ is the subset of all individual records that belong to household $h^t(i)$. When comparing $\boldsymbol{A}$ to $\boldsymbol{B}$, we are essentially comparing the group of all individuals from household $\boldsymbol{A}$ to household $\boldsymbol{B}$. The first step is to calculate the similarity across all pairs of individuals. As previously mentioned, this similarity can be calculated in various ways and is typically a consolidation of the individual pairwise field similarities (see Section 5.2.4). The similarity scores are shown visually on the edges of the left graph in Fig. 5.1. Then we set an arbitrary cutoff that determines which pairs are matches or non-matches. The matches, who have a higher similarity than our cutoff of 0.5, are shown in the right of Fig. 5.1. Now that we have the similarity scores, we can calculate the number of matching pairs. Because we set this cutoff at 0.5, only the Patricks and the Ellens match meaning that $|\hat{\boldsymbol{m}}| = 2$. Additionally, only the Patrick / Patrick and Ellen / Ellen similarities get to contribute to the numerator of their group similarity. We can now proceed to calculate the group similarity as shown in 5.17.

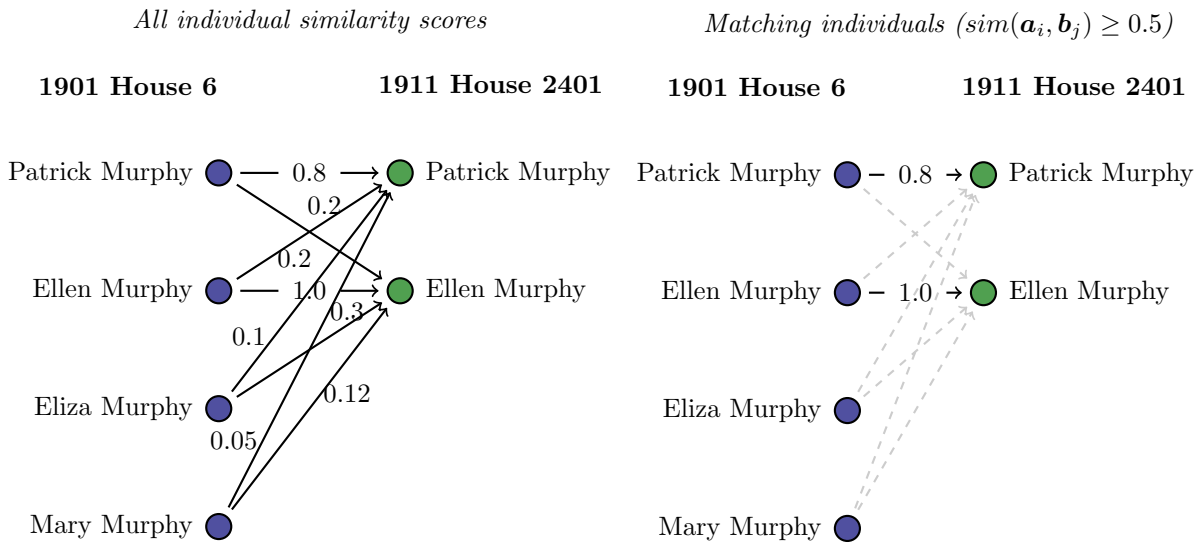


Figure 5.1: Bipartite graphs of the 6th household in 1901 and the 2401st household in 1911. Similarity scores are shown for the pairs of individuals. On the right we only keep the similarity for those pairs that are considered a match (similarity score greater than or equal to 0.5.)

$$\boldsymbol{A} = \mathbb{X}^{1901}(h_6) = \{\boldsymbol{X}^{1901}_{32.}, \boldsymbol{X}^{1901}_{33.}, \boldsymbol{X}^{1901}_{34.}, \boldsymbol{X}^{1901}_{35.}\}$$

$$\boldsymbol{B} = \mathbb{X}^{1911}(h_{2401}) = \{\boldsymbol{X}^{1911}_{9596.}, \boldsymbol{X}^{1911}_{9597.}\}$$

$$\hat{\boldsymbol{m}} = \{(\boldsymbol{X}^{1901}_{32.}, \boldsymbol{X}^{1911}_{9596.}), (\boldsymbol{X}^{1901}_{33.}, \boldsymbol{X}^{1911}_{9597.})\}$$

$$sim_{\text{group-linkage}}(\boldsymbol{A}, \boldsymbol{B}) = \frac{\sum\limits_{(\boldsymbol{a}_i, \boldsymbol{b}_j) \in \hat{\boldsymbol{m}}} sim(\boldsymbol{a}_i, \boldsymbol{b}_j)}{|\boldsymbol{A}| + |\boldsymbol{B}| - |\hat{\boldsymbol{m}}|} = \frac{0.8 + 1.0}{4 + 2 - 2} = \frac{1.8}{4} = 0.45$$

Using the information in Fig. 5.1 we were able to calculate the group linkage similarity between 1901 House 6 and 1911 House 2401 (5.17). But in practice, sometimes we want to compare 1901 House 6 to

multiple potential 1911 household matches to determine which household has the highest similarity. For example, in the paper where group linkage similarity is introduced ([43]), they compare multiple authors (that each have various sets of citations) to determine which authors are actually the same individual. [‡] We demonstrate the [43] group linkage application in Fig. 5.2 and Fig. 5.3. We already introduced Fig. 5.2 where we compare 1901 House 6 to 1911 House 2401 and calculate a group linkage similarity of 0.45. However, we also want to see how similar House 6 is to 1911 House 81 to see if House 81 may be a better match. In Fig. 5.3 we visualize and calculate a group linkage similarity of 0.116. Therefore, they would classify House 2401 to be the best match for House 6 because it has a higher group similarity.

In the historical record linkage context, Fu, Christen, and Boot utilize the existing group linkage measure as a post-processing step to help resolve intransitivities among individual records [23]. They first link individuals with the expectation of some erroneous links, and then use existing household structure to refine and improve the individual linkage results. If an individual from the first data set is linked to two individuals within the second data set, they use the household similarity to determine which of the two potential matches is chosen. [23] would, in practice, utilize the group similarity slightly differently than [43] because [23] is concerned with linking individuals and [43] is concerned with linking the group. [23] use household group similarity to help determine the correct individual-level match in the case of multiple matching individuals. We see in our example that, Ellen Murphy from 1901 House 6 matches to Ellen Murphy (at a threshold above 0.5) in both 1911 Houses: 2401 and 81. Therefore, following the approach in [23] we would choose the 1911 Ellen whose group similarity is higher (in this case that would be the first Ellen). If one did not want household information to influence the individual linkage, there are large benefits to using group linkage as a post-processing step. [§] One potential downside of group linkage is its dependence on the strength of the individual links. Confident individual links will likely lead to a dependable group linkage measure, but individual links are not always strong. If individual similarities are used in the group linkage measure, it may be unwise to use the group linkage measure in a secondary individual level linkage because of potential overfitting.

[‡]They first use individual based matching techniques (e.g., cosine similarity) to assign a similarity score to pairs of citations. They then arbitrarily threshold these similarity scores, determining that those above the threshold are matches. They use the individual citation matches within an author pair to determine the group linkage score of the two author groups. More specifically, they take the bipartite graph (left is a group of an author's citations from one database, right is a group of an author's citations from a second database) and take an average of the weights of matching elements across the two groups.

[§]Sometimes there are government mandates / regulation that prevents additional household information from being used in the individual linkage. Luiza Antonie (committee member) informed us that this is the case in some historical Canadian linkages.

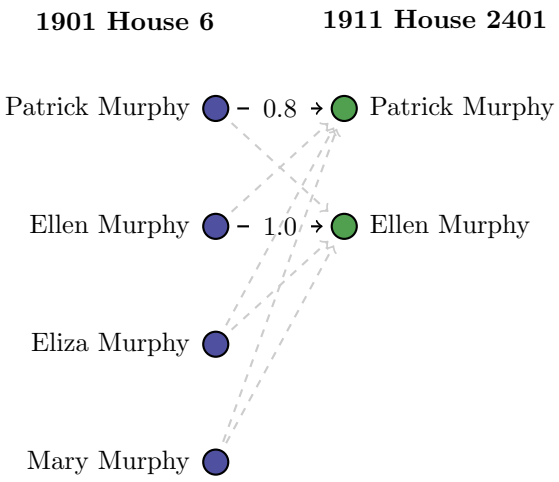


Figure 5.2: Potential house match (1911 House 2401) with group linkage similarity of 1.8 / 4 = 0.45

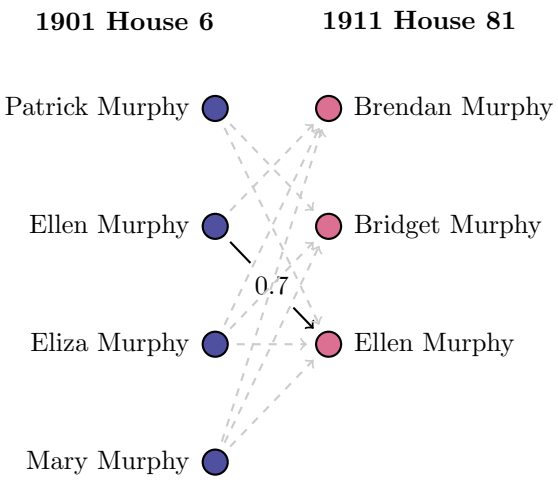


Figure 5.3: Potential house match (1911 House 81) with group linkage similarity of 0.7 / 6 = 0.116

Both papers utilize the same group linkage similarity score, but in different ways based on the focus to link either individuals or groups. We have demonstrated both approaches using historical Irish Census records in Fig. 5.2 and Fig. 5.3. In this thesis we use group linkage as one way to calculate a similarity score for two households in their entirety. We will discuss how we use this metric in later sections.

5.4 Comparing Similarity Measures

5.4.1 Equality of Group Linkage and Jaccard Similarities

We show that the group linkage with a similarity function that indicates whether two individuals match is the same as the Jaccard Index for identifying unique individuals. We show this at both the individual (Theorem 5.1) and field/set level (Theorem 5.2).

In Theorem 5.1 we show that the Group Linkage of household $\boldsymbol{A}$ and household $\boldsymbol{B}$ is the same as the Jaccard of these households. Note that in Equation 5.7 for Jaccard similarity we originally compared sets of a specific field (e.g. first name). $\mathbb{A} \cap \mathbb{B}$ represented the field values that appeared in both set $\mathbb{A}$ and set $\mathbb{B}$ where $\mathbb{A} \cup \mathbb{B}$ additionally represented those field values that appeared in only set $\mathbb{A}$ or set $\mathbb{B}$ but not the other. When thinking about the actual individuals (instead of their fields) we can consider $\boldsymbol{A} \cap \boldsymbol{B}$ to

$\boldsymbol{a}_i$	$\boldsymbol{b}_j$	$sim(\boldsymbol{a}_i, \boldsymbol{b}_j)$
Patrick Murphy	Patrick Murphy	1
Ellen Murphy	Ellen Murphy	1
Eliza Murphy	Patrick Murphy	0
Mary Murphy	Ellen Murphy	0
Patrick Murphy	Ellen Murphy	0
Ellen Murphy	Patrick Murphy	0
Eliza Murphy	Ellen Murphy	0
Mary Murphy	Patrick Murphy	0

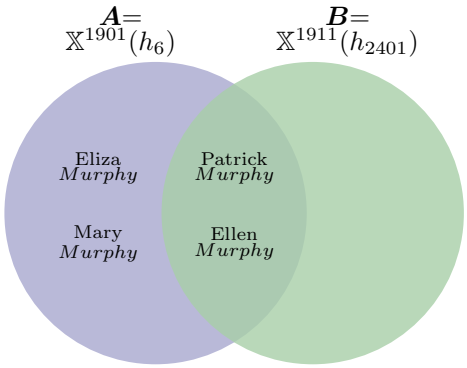


Figure 5.4: Visualizing the relationship between group linkage and Jaccard similarities for two households, $\boldsymbol{A}$ and $\boldsymbol{B}$. The 1901/1911 Patricks and Marys are the same person but Ellen and Eliza from 1901 have no match in 1911 and therefore belong outside of the union of the two households.

represent the individuals who had records in both 1901 and 1911 (and therefore their unique entity should only be counted once) and $\boldsymbol{A} \cup \boldsymbol{B}$ to additionally include those individuals who only appeared in one of the two census record databases. $|\boldsymbol{A} \cup \boldsymbol{B}|$ gives us the total number of unique individuals that appear in either 1901 or 1911. In Figure 5.4 we present a visual representation of what is happening. We first assume we have the similarities between pairs of records and therefore have an existing match status. The exact match status of these records is often unknown (but can be estimated, for example using $sim(\boldsymbol{a}_i, \boldsymbol{b}_j)$). To get this match status we could calculate the average field similarity or the cosine similarity of pairs of individuals and then apply a binary threshold to those similarities. Alternatively we could build a (unsupervised or supservised) statistical model and threshold the output match probabilities. Regardless, the group linkage similarity requires a set of starting matches (the pairs that will form one entity within $\hat{\boldsymbol{m}}$). Once we have these similarities we can calculate both the group linkage and Jaccard similarities. In the right of Figure 5.4 we see the matches within the union whereas the individuals without a match are outside of the union. We will show that under specific similarity conditions, the group linkage and the Jaccard are the same.

We can also show that the group linkage similarity of two sets ($\mathbb{A}$ and $\mathbb{B}$) is the same as the Jaccard similarity of those sets, once again provided that the similarity is 1 for exact matching string pairs and 0 for non-matching string pairs. When thinking about group similarity at the field level (as opposed to the individual level) we consider $\hat{m}$ to be the field string pairs that match across the two groups (instead of $\hat{\boldsymbol{m}}$ representing the individual pairs that match). In Figure 5.5 we show the similarity scores as well as the resulting Venn diagram of the strings. Those that match belong in the union and those individuals without a match are not in the union.

a_i	b_j	$sim(a_i, b_j)$
"Patrick"	"Patrick"	1
"Ellen"	"Ellen"	1
"Eliza"	"Patrick"	0
"Mary"	"Ellen"	0
"Patrick"	"Ellen"	0
"Ellen"	"Patrick"	0
"Eliza"	"Ellen"	0
"Mary"	"Patrick"	0

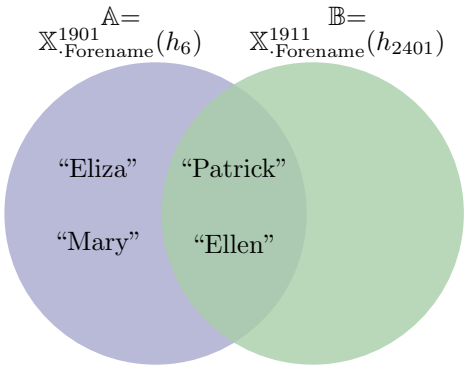


Figure 5.5: Visualizing the relationship between group linkage and Jaccard similarities for the sets of first names ($\mathbb{A}$ and $\mathbb{B}$) across the two households. "Patrick" and "Mary" appears in both 1901 and 1911 but the names "Mary" and "Eliza" do not appear in 1911.

Theorem 5.1 (Equality of Group Linkage and Jaccard for individuals)**.** *The group linkage similarity of two groups ($\boldsymbol{A}$ and $\boldsymbol{B}$) is the same as the Jaccard similarity of the two groups, provided we have an initial set of (likely estimated) matches between individuals. Specifically, $sim(\boldsymbol{a}_i, \boldsymbol{b}_j) = 1$ if $\boldsymbol{a}_i$ is considered to be the same entity/person as $\boldsymbol{b}_j$ and $sim(\boldsymbol{a}_i, \boldsymbol{b}_j) = 0$ if they are considered to be different entities.*

$$sim_{group\text{-}linkage}(\boldsymbol{A}, \boldsymbol{B}) = Jaccard(\boldsymbol{A}, \boldsymbol{B}) \tag{5.17}$$
$$if\ sim(\boldsymbol{a}_i, \boldsymbol{b}_j) = \begin{cases} 1 & if\ \boldsymbol{a}_i \hat{\sim} \boldsymbol{b}_j \\ 0 & if\ \boldsymbol{a}_i \hat{\nsim} \boldsymbol{b}_j \end{cases}$$

Proof of Theorem.

$$\boldsymbol{A} = \{\boldsymbol{a}_{1,\cdot}, \boldsymbol{a}_2, ..., \boldsymbol{a}_{n_{h^{1901}(i)}}\}$$
$$\boldsymbol{B} = \{\boldsymbol{b}_{1,\cdot}, \boldsymbol{b}_2, \boldsymbol{b}_3, ..., \boldsymbol{b}_{n_{h^{1911}(j)}}\}$$
$$\hat{\boldsymbol{m}} = \{(\boldsymbol{a}_i, \boldsymbol{b}_j) \in \boldsymbol{A} \times \boldsymbol{B} : \boldsymbol{a}_i \hat{\sim} \boldsymbol{b}_j\}$$

$$
\begin{aligned}
sim_{\text{group-linkage}}(\boldsymbol{A}, \boldsymbol{B}) &= \frac{\sum_{(\boldsymbol{a}_i, \boldsymbol{b}_j) \in \hat{\boldsymbol{m}}} sim(\boldsymbol{a}_i, \boldsymbol{b}_j)}{|\boldsymbol{A}| + |\boldsymbol{B}| - |\hat{\boldsymbol{m}}|} \\
&= \frac{\sum_{(\boldsymbol{a}_i, \boldsymbol{b}_j) \in \hat{\boldsymbol{m}}} 1}{|\boldsymbol{A}| + |\boldsymbol{B}| - |\hat{\boldsymbol{m}}|} \\
&= \frac{|\hat{\boldsymbol{m}}|}{|\boldsymbol{A}| + |\boldsymbol{B}| - |\hat{\boldsymbol{m}}|} \\
&= \frac{|\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A}| + |\boldsymbol{B}| - |\boldsymbol{A} \cap \boldsymbol{B}|} \\
&= \frac{|\boldsymbol{A} \cap \boldsymbol{B}|}{|\boldsymbol{A} \cup \boldsymbol{B}|} \\
&= \text{Jaccard}(\boldsymbol{A}, \boldsymbol{B})
\end{aligned}
$$

□

Theorem 5.2 (Equality of Group Linkage and Jaccard for field sets)**.** *We show that group linkage similarity of two sets ($\mathbb{A}$ and $\mathbb{B}$) is the same as the Jaccard similarity of those sets, given that we use an exact match string similarity to determine the matches. Specifically, $sim(a_i, b_j) = 1$ if a_i is the same string as b_j and $sim(a_i, b_j) = 0$ if the strings do not match.The strings from $\mathbb{A}$ that have a match in $\mathbb{B}$ belong in the union and those that don't appear in the other household belong within their original respective household (but not in the union). The union and m are synonymous in that the strings that appear in both households make up m which is the set of all matching string pairs.*

$$
\begin{aligned}
&\textit{Jaccard}(\mathbb{A}, \mathbb{B}) = sim_{\textit{group-linkage}}(\mathbb{A}, \mathbb{B}) \\
&\textit{if } sim(a_i, b_j) = \begin{cases} 1 & \textit{if } a_i \hat{\sim} b_j \\ 0 & \textit{if } a_i \hat{\nsim} b_j \end{cases} \\
&\textit{if } m = \{(a_i, b_j) \,:\, a_i \hat{\sim} b_j\}
\end{aligned}
$$

Proof of Theorem.

$$\text{Let } \mathbb{A} = \{a_1, a_2, ..., a_{n_a}\}$$
$$\text{Let } \mathbb{B} = \{b_1, b_2, ..., b_{n_b}\}$$
$$m = \{(a_i, b_j) \,:\, a_i \hat{\sim} b_j\}$$

To simplify notation we let $d_{S_{\mathbb{A}}}(x) = d_{Supp(\mathbb{A})}(x)$.

$$\begin{aligned}
\text{Jaccard}(\mathbb{A}, \mathbb{B}) &= \frac{\sum\limits_{x \in \mathbb{A} \cup \mathbb{B}} \min\left(d_{S_{\mathbb{A}}}(x), d_{S_{\mathbb{B}}}(x)\right)}{\sum\limits_{x \in \mathbb{A} \cup \mathbb{B}} \max\left(d_{S_{\mathbb{A}}}(x), d_{S_{\mathbb{B}}}(x)\right)} \\
&= \frac{\sum\limits_{x \in \mathbb{A} \cap \mathbb{B}} \min\left(d_{S_{\mathbb{A}}}(x), d_{S_{\mathbb{B}}}(x)\right) + \sum\limits_{x \in \mathbb{A} \cap \mathbb{B}^c} \min\left(d_{S_{\mathbb{A}}}(x), d_{S_{\mathbb{B}}}(x)\right) + \sum\limits_{x \in \mathbb{B} \cap \mathbb{A}^c} \min\left(d_{S_{\mathbb{A}}}(x), d_{S_{\mathbb{B}}}(x)\right)}{\sum\limits_{x \in \mathbb{A} \cap \mathbb{B}} \max\left(d_{S_{\mathbb{A}}}(x), d_{S_{\mathbb{B}}}(x)\right) + \sum\limits_{x \in \mathbb{A} \cap \mathbb{B}^c} \max\left(d_{S_{\mathbb{A}}}(x), d_{S_{\mathbb{B}}}(x)\right) + \sum\limits_{x \in \mathbb{B} \cap \mathbb{A}^c} \max\left(d_{S_{\mathbb{A}}}(x), d_{S_{\mathbb{B}}}(x)\right)} \\
&= \frac{\sum\limits_{x \in \mathbb{A} \cap \mathbb{B}} 1 + \sum\limits_{x \in \mathbb{A} \cap \mathbb{B}^c} 0 + \sum\limits_{x \in \mathbb{B} \cap \mathbb{A}^c} 0}{\sum\limits_{x \in \mathbb{A} \cap \mathbb{B}} 1 + \sum\limits_{x \in \mathbb{A} \cap \mathbb{B}^c} 1 + \sum\limits_{x \in \mathbb{B} \cap \mathbb{A}^c} 1} \\
&= \frac{\sum\limits_{(a_i, b_j) \in m} 1}{|m| + |\mathbb{A} \cap \mathbb{B}^c| + |\mathbb{B} \cap \mathbb{A}^c|} \\
&= \frac{\sum\limits_{(a_i, b_j) \in m} 1}{|m| + |\mathbb{A}| - |m| + |\mathbb{B}| - |m|} \\
&= \frac{\sum\limits_{(a_i, b_j) \in m} 1}{|\mathbb{A}| + |\mathbb{B}| - |m|} \\
&= sim_{\text{group-linkage}}(\mathbb{A}, \mathbb{B})
\end{aligned}$$

□

5.5 Chapter Conclusion

In this Chapter we first introduce mathematical notation for use throughout the remaining chapters. We introduce approaches for comparing fields, individual records, groups of fields, and groups of individual records. We re-write the Jaccard and the Ruzicka Jaccard so that they can easily be compared, especially in the case of multisets. We can show the equality between the Jaccard and the Group Linkage similarities, under conditions about the group linkage similarity score, although in the remainder of this thesis we will

use the Jaccards and the Group Linkage within slightly different contexts. We use the Jaccard to explore sets of group fields while we use group linkage to compare two households in their entirety. The group linkage similarity is heavily dependent on the provided similarity scores and the resulting match status of the individuals or fields across households.

Chapter 6

Supervised Models

In this section we explore classification methods for record linkage. Within record linkage, we typically try to predict whether two individuals are a match (binary yes/no) from information (often similarity scores) about the record pair. Within this setting, a row of data represents a *pair* of individuals, one from each data source, (e.g. one individual from 1901 and one from 1911 for the Ireland census problem). If we happen to know the match status of a set of record pairs, we can build supervised models, which will be the focus of this chapter. Because we know y_{ij} (i.e., whether or not the two individuals match) we can model y_{ij} as a function of the pair's features/covariates. These covariates (X_{ij}) are typically similarity scores comparing the two individuals (e.g., name similarity, geographic distance). It is typical to denote the relationship between the probability of a pair matching ($y_{ij} = 1$) and the covariates (X_{ij}) by the following classification equation:

$$\Pr(y_{ij} = 1) = f(X_{ij}). \tag{6.1}$$

Please note that we can use any classification model, such as logistic regression (Equation 6.2), general additive models (GAMs) (Equation 6.3), or random forests to classify our record pairs [62] [27] [30]. Logistic regression is commonly used in record linkage due to its interpretability; we chose to include GAMs for their ease at modeling non-linear variables; and random forests have been shown to achieve strong predictive performance in this field [57] [60]. In the context of this thesis we are less concerned with finding the machine learning model that performs best and more concerned with exploring and understanding how changes to the underlying model structure impact linkage (specifically with respect to certain sub-populations).

$$\Pr(y_{ij} = 1) = \text{logit}^{-1}(X_{ij}\beta) \tag{6.2}$$

$$\Pr(y_{ij} = 1) = \text{logit}^{-1}(\beta_0 + f_1(X_{ij1}) + \cdots + f_k(X_{ijk})) \tag{6.3}$$

This chapter proposes adaptions of classification models for record linkage that incorporate household information in various ways. In Sections 6.1 and 6.2 we prepare the individual and household data for classification. In section 6.3 we formally state the models (with their adaptations). We discuss model validation and our training/testing splits within Section 6.4. In Section 6.5 we report performance results and analyze these differences. We briefly conclude in Section 6.6.

6.1 Features from Individuals to Individual Comparisons

In Table 6.1 we see the original two record databases side by side. In order to compare these records, one must process the data to create the covariates X_{ij} for the record linkage model.In Section 5.2.4 we demonstrated how to consolidate pairwise similarities into a vector of similarities, which is a common pre-processing step. The visual representation of what this looks like for the entire data set is shown here in Table 6.2.

		1901					***1911***		
ID	**Forename**	**Surname**	$\cdots$	**Sex**	**ID**	**Forename**	**Surname**	$\cdots$	**Sex**
1	Maryanne	Sheridan	$\cdots$	Female	1	Maryanne	Waldron	$\cdots$	Female
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
n_{1901}	Paul	Hanlon	$\cdots$	Male	n_{1911}	Matilda	Moore	$\cdots$	Female

Table 6.1: Original Census Records from 1901 and 1911.

More specifically, Table 6.2 shows the full matrix that is created when we calculate comparisons from the original records, where the rows are now associated with the pairwise comparison of two individuals. As previously mentioned, we sometimes know whether or not two individuals are the same person (shown via the column "Match" in Table 6.2). In this application, we will know the match status for individuals who have been hand labeled via our interface (as described in Chapter 2).

6.2 Features from Households to Household Comparisons

Although it is critical to compare records at the individual level, as we've discussed and shown anecdotally, it can be additionally useful to use household information to help label records. Similar to how we compare individuals above, household data also needs to be prepared for record linkage. We use the tools presented in Section 5.3, i.e. Jaccard Similarity, Ruzicka Jaccard Similarity, Group Linkage Similarity, to produce a

Individual Comparison Data

		$\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}$	$sim\left(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911}\right)$			
1901 ID (i)	1911 ID (j)	Match	JW(Forename)	JW(Surname)	$\cdots$	Exact(Sex)
1	1	1	1	0.49	$\cdots$	1
1	2	NA	0	0.22	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
n_{1901}	n_{1911}	0	0.6	0.45	$\cdots$	0

Table 6.2: Comparison/similarity data across 1901 and 1911 individual records.

similarity matrix for the households. We subset the individual level data by each household and then compare the sets of individuals across 1901 and 1911 according to their household ID. Using the group-level similarity metrics described in Section 5.3, we assign household-level similarity scores (covariates) to Table 6.3. Note $h^{1911}(1) = h^{1911}(2) = h^{1911}(3) = h^{1911}(4)$ because the first four individuals in 1911 all belong to the same household.

Household Comparison Data

		$\boldsymbol{X}^{1901}(h^{1901}(i)) \sim \boldsymbol{X}^{1911}(h^{1911}(j))$	$sim\left(\boldsymbol{X}^{1901}(h^{1901}(i)), \boldsymbol{X}^{1911}(h^{1911}(j))\right)$			
1901 HouseID	1911 HouseID	Match	*Jaccard* (Forename)	*Jaccard* (Surname)	$\cdots$	Group Linkage
$h^{1901}(1)$	$h^{1911}(1)$	NA	0.7	1	$\cdots$	0.85
$h^{1901}(1)$	$h^{1911}(5)$	0	0.1	0	$\cdots$	0.15
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$h^{1901}(n_{1901})$	$h^{1911}(n_{1911})$	1	1	1	$\cdots$	1

Table 6.3: Comparison/similarity data across 1901 and 1911 households.

In the following sections (6.3) we will discuss how we incorporate this new household similarity matrix into the record linkage process.

6.3 Classification Models

We take a standard approach to classifying pairs of records as matches or non-matches, by building models that output the probability of matching for two individuals. We can then threshold this output to determine our final classes. In this section we outline all of our models for building classification models. We first start with a baseline model of just individual-specific features (Section 6.3.1) and then adapt this model to incorporate household information. We can incorporate household information directly into the model using household covariates like the Ruzicka Jaccard similarities (Equation 6.6) or by using a household similarity

derived from individual covariates like the Group Linkage Similarity (Equation 6.7). We can also utilize a Group Linkage Similarity that has been derived from estimated individual pair probabilities (Section 6.3.3). Finally, we use multilevel models that incorporate the structure of individuals within households.

6.3.1 Baseline Model

We are interested in predicting whether a pair of individuals match from multiple covariates representing the similarity of the pair. In this classification setting (adapted from Equation 6.1), y_{ij} represents whether or not the *pair* of individuals $(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911})$ match ($y_{ij} = 1$ or $\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}$ indicate a match, $y_{ij} = 0$ or $\boldsymbol{X}_i^{1901} \nsim \boldsymbol{X}_j^{1911}$ indicate the two individuals do not match). X represents the covariates about the pair of individuals (i.e., $sim(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911})$). We denote this below:

$$y_{ij} = \begin{cases} 0 & \text{if } \boldsymbol{X}_i^{1901} \nsim \boldsymbol{X}_j^{1911} \\ 1 & \text{if } \boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911} \end{cases}$$
$$X_{ij} = sim(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911}).$$

As a reminder from our notation section (Section 5.1), $\boldsymbol{X}_i^{1901}$ is the ith record in the 1901 census database and $\boldsymbol{X}_j^{1911}$ is the jth record in the 1911 census database. We do not necessarily have to build a model on all i and all j from the data (blocking enforces that we do *not* compare all i and j). We will discuss splitting the data into training / testing later in Section 6.4. We predict whether i all j match from attributes about the similarity of the pair of records: $sim(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911})$. The similarity between the two individual records $(sim(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911}))$ is typically a vector of similarity scores about the pair. For example, the vector could be the first name similarity, last name similarity, and the sex similarity. Using the above notation we can abstractly define the classification model in the following equation:

$$\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}) = f(sim(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911})). \tag{6.4}$$

In its simplest form we can build Model 6.4 from some of the comparison data in Table 6.2 (that is, with individual pairwise similarity scores). One example of a model we could build: logistic regression using the Jaro-Winkler similarity of first names, last names, location, and birthplace, whether the gender is an exact match, and the absolute difference in ages is shown in here:

$$\begin{aligned}\text{logit}(\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911})) =& \beta_0 + sim_{\text{jaro-winkler}}\,(X_{i,Forename}^{1901},\ X_{j,Forename}^{1911})\beta_1 + \\ & sim_{\text{jaro-winkler}}\,(X_{i,Surname}^{1901},\ X_{j,Surname}^{1911})\beta_2 + \\ & sim_{\text{exact}}\,(X_{i,Sex}^{1901},\ X_{j,Sex}^{1911})\beta_3 \ +\ sim_{\text{jaro-winkler}}\,(X_{i,Location}^{1901},\ X_{j,Location}^{1911})\beta_4, \ + \\ & sim_{\text{abs-diff}}\,(X_{i,Age}^{1901},\ X_{j,Age}^{1911})\beta_5 \ +\ sim_{\text{jaro-winkler}}\,(X_{i,Birthplace}^{1901},\ X_{j,Birthplace}^{1911})\beta_6.\end{aligned}$$

The set of covariates chosen in a record linkage model will change depending on context, type of model used, performance etc. In our paper [21], we chose these variables to build our baseline model largely because these fields have little missing data across both years. Additionally, in [21] we predominantly found that random forests performed best out-of-sample.

6.3.2 Adding Household Covariates

Once we have calculated similarity scores at the household level (Section 6.2) we can incorporate those similarities into our existing record linkage classification models. As a reminder of notation (Section 5.1), $\boldsymbol{X}^t(h^t(i)) \subset \boldsymbol{X}^t = \{\boldsymbol{X}_{i\cdot}^t : i \in h^t(i)\}$. This means that $\boldsymbol{X}^t(h^t(i))$ is all records from time t that were recorded in house $h^t(i)$. So below, $\boldsymbol{X}^{1901}(h^{1901}(i))$ represents all individuals from the same household as individual i and $\boldsymbol{X}^{1911}(h^{1911}(j))$ represents all individuals who share a household with individual j. We might define this type of classification model with added household covariates as:

$$\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}) = f\big(sim(\boldsymbol{X}_i^{1901},\ \boldsymbol{X}_j^{1911}),\ sim(\boldsymbol{X}^{1901}(h^{1901}(i)),\ \boldsymbol{X}^{1911}(h^{1911}(j)))\big). \tag{6.5}$$

In Section 5.3 discussed two main ways of calculating household similarity: (Ruzicka) Jaccard (Section 5.3.3) and group linkage similarity (Section 5.3.4). Because calculating the Jaccard similarity is an unsupervised process (it is calculated directly from the original records) it can easily be calculated and included in a classification model. In Equation 6.6, we add in covariates for various example fields at the household-level, shown here:

$$\begin{aligned}\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}) = f\big(& sim_{\text{jaro-winkler}}\,(X_{i,Forename}^{1901},\ X_{j,Forename}^{1911}),\ \cdots, \\ & sim_{\text{jaro-winkler}}\,(X_{i,Birthplace}^{1901},\ X_{j,Birthplace}^{1911}), \\ & \text{Jaccard}_{\text{ Ruzicka}}\,(\mathbb{X}_{\cdot\text{Forename}}^{1901}(h^{1901}(i)),\ \mathbb{X}_{\cdot\text{Forename}}^{1911}(h^{1911}(j))),\ \cdots, \\ & \text{Jaccard}_{\text{ Ruzicka}}\,(\mathbb{X}_{\cdot\text{Birthplace}}^{1901}(h^{1901}(i)),\ \mathbb{X}_{\cdot\text{Birthplace}}^{1911}(h^{1911}(j)))\big).\end{aligned} \tag{6.6}$$

This example utilizes the Ruzicka Jaccard but the original Jaccard could also be used. We build the model using similarity metrics about both the individual and their household. When predicting whether 1901 person i is the same as 1911 person j, we model the similarity scores for the two individuals as well as the similarity scores for their two households.

Instead of comparing households field by field, we might want one metric that describes the overall similarity of two households. In Section 5.3.4 we introuced the group linkage similarity , which does just that. Calculating the group linkage measure, however, requires individual pairwise similarity scores. If we can calculate those in an unsupervised fashion (e.g., calculating the average similarity of the Jaro-Winkler scores across individuals), we can incorporate it directly into the individual-level model like we did with the Jaccard variables. Below, in Equation 6.7 we show an example of what this looks like:

$$\begin{aligned}\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}) = f\big(&sim_{\text{jaro-winkler}}\,(X_{i,Forename}^{1901},\ X_{j,Forename}^{1911}),\ \cdots,\\ &sim_{\text{jaro-winkler}}\,(X_{i,Birthplace}^{1901},\ X_{j,Birthplace}^{1911}),\\ &sim_{\text{group-linkage}}(\boldsymbol{X}^{1901}(h^{1901}(i)),\ \boldsymbol{X}^{1911}(h^{1911}(j)))\big). \end{aligned} \tag{6.7}$$

6.3.3 Multi-Stage Group Linkage Model

As done in [23], we can calculate the group linkage similarity using the individual pairwise predicted probabilities of matching. From there, we can use that group linkage similarity as a feature in a secondary classification model. We would perform the following steps in Alorithm 1 to complete the process.

Algorithm 1: Multi-stage modeling of individual records using group linkage similarity

Input: All records from 1901 and 1911 ($\boldsymbol{X}^{1901}$ and $\boldsymbol{X}^{1911}$)
Household IDs that map i to $h^{1901}(i)$ and j to $h^{1911}(j)$
Pairwise match cutoff value c

Output: A model that predicts the probability of matching for pairs of $\boldsymbol{X}^{1901}$, $\boldsymbol{X}^{1911}$

1 Let $\mathcal{S}_1^{1901}, \mathcal{S}_2^{1901}$ be subsets of $\{1, ..., n_{1901}\}$ and $\mathcal{S}_1^{1901} \cap \mathcal{S}_2^{1901} = \emptyset$;
2 Let $\mathcal{S}_1^{1911}, \mathcal{S}_2^{1911}$ be subsets of $\{1, ..., n_{1911}\}$ and $\mathcal{S}_1^{1911} \cap \mathcal{S}_2^{1911} = \emptyset$;
3 Build a model for $\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911})$ with Equation 6.4 using $i \in \mathcal{S}_1^{1901}, j \in \mathcal{S}_1^{1911}$;
4 Calculate $sim_{\text{group-linkage}}(\boldsymbol{X}^{1901}(h^{1901}(i)),\ \boldsymbol{X}^{1911}(h^{1911}(j)))$ for household pairs of $i \in \mathcal{S}_2^{1901}, j \in \mathcal{S}_2^{1911}$ using Equation 5.16. Let $sim(\boldsymbol{a}_i, \boldsymbol{b}_j)$ be the predicted probabilities from the model for $\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911})$ on Line 3 ;
5 Build a second model for $\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911})$ with Equation 6.7 using $i \in \mathcal{S}_2^{1901}, j \in \mathcal{S}_2^{1911}$ and the $sim_{\text{group-linkage}}$ calculated previously in Line 4 ;
6 Using the model from Line 5, predict the probability of matching for holdout data and determine links using cutoff c ;

Using this multi-stage model we first build a model to calculate group linkage similarity and then utilize that (potentially better household similarity approximation) in a secondary individual model. We can predict on any data that was not seen in either of the first two models. One challenge of this model is the need for multiple holdout data sets. In practice this is a limitation because each segment of the algorithm uses less data than an algorithm that doesn't build two models.

6.3.4 Hierarchical Classification Models

We have shown how we can incorporate similarities about the two individuals' households directly into the model, but we are not accounting for variation that occurs at the household level [24]. We instead could fit a model where each household pair variables receives its own covariate in an individual pairwise model. We first write the most general form of a hierarchical linear model in Equation 6.8, shown here:

$$\begin{aligned} \Pr(y_{ij} = 1) &= \text{logit}^{-1}(X_{ij}\beta + \alpha_{h_{ij}}), \text{ for } y = 1, \ldots, n \\ \alpha_{h_{ij}} &\sim N(U_{h_{ij}}\,\gamma,\ \sigma_\alpha^2) \text{ for } h_{ij} = 1, \ldots, w. \end{aligned} \tag{6.8}$$

We are still modeling the probability of y from X but we are allowing the coefficient α to vary by the group h. Adapting this to the record linkage setting we replace the covariate matrix X_{ij} with the pairwise similarities $sim(\boldsymbol{X}_i^{1901},\ \boldsymbol{X}_j^{1911})$. You can see that $\alpha_{h^{1901}(i)h^{1911}(j)}$ is the coefficient specific to the household *pair* $(h^{1901}(i),\ h^{1911}(j))$. Any pair that originated from households $h^{1901}(i)$ and $h^{1911}(j)$ will have the same α value. The multilevel model, updated in our specific application, is shown here:

$$\begin{aligned} \Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}) &= \text{logit}^{-1}(sim(\boldsymbol{X}_i^{1901},\ \boldsymbol{X}_j^{1911})\beta + \alpha_{h^{1901}(i)h^{1911}(j)}), \\ &\quad \text{for } i = 1, \ldots, n_{1901},\ j = 1, ..., n_{1911} \\ \alpha_{h^{1901}(i)h^{1911}(j)} &\sim N(U_{h^{1901}(i)h^{1911}(j)}\,\gamma,\ \sigma^2_{\alpha_{h^{1901}(i)h^{1911}(j)}}), \\ &\quad \text{for } a = 1, ..., n_{h^{1901}},\ b = 1, ..., n_{h^{1911}}. \end{aligned} \tag{6.9}$$

6.4 Model Validation

Model evaluation typically follows model building. There is commonly an extremely large class imbalance in record linkage, because there are many non-matching pairs and few matching pairs. Therefore, evaluation metrics that include true negatives are typically not used because models can inadvertently appear stronger than they actually are[29]. For example, overall accuracy about whether a pair was correctly labeled as a

match or a non-match would still be high if we naively classified all pairs as non-matches. Please also note that we do not apply any techniques to address the class imbalance directly. But, given the imbalance, it is more common to examine the precision and recall of record linkage models. It is also common to combine precision and recall into summary measures like the F-Score, but this metric has been recently criticized and is no longer recommended within record linkage[25]. When reporting evaluation results, we can use precision recall curves to visualize model performance under multiple cutoff values simultaneously; this is important when we don't know the "correct" cutoff. The area under the precision recall curve (AUC) is a summary statistic combining both precision and recall, evaluated at these different cutoff values. Typically, the higher the AUC, the better the classifier. Despite the flaws of the AUC, noted by Hand [26], use of AUC is still commonplace and easily interpretable without having to select a cutoff value and therefore we report both precision and recall as well as the AUC when evaluating our models.

It is also important to evaluate our models fairly on testing data that has not been seen by a model. Appropriately training, validating, and testing models is critical to the model selection and validation process, especially when determining what models perform "best". In our application, we are less concerned with finding the "best" model and more with understanding model differences. We do not suggest that one model or one set of modeling steps should be used in future analyses but instead argue that attention should be paid to the full data collection and modeling pipeline. For these reasons, we use a train/test split that splits the data at the reference record level. This means that of all unique reference records, 60% will be in the training data and 40% will be in the testing set. In the training set, this equates to around $1,000,000$ rows without blocking and $8,000,000$ million rows with blocking. Within testing this is about $680,000$ without blocking and $5,200,000$ with. To reiterate, we could have left a percentage of the data completely unseen to compare all eventual models as a final step but argue that our goal is not to make that final comparison and declare a "best" model.

Our main concern with splitting our data by reference record is that a single household pair could get split across training and testing. But, we would be unable to build a hierarchical model for households / individuals if we split by the household level. We do build some non-hierarchical models where we split by the reference record's household, to compare results with the reference split performance. A potential future area of research would include utilizing different data splits. Because our data represents pairs of individuals or pairs of households, splitting the data is more complicated because even if we split by unique reference/candidate household IDs, both the reference and candidate households are often compared to numerous other households. In an ideal world, the training and testing populations would look very similar in distribution but there would not be any overlap between indidivuals or households across training and testing. Perhaps pre-clustering the data could help us achieve this result in a rigorous way, but we leave that for future work.

6.5 Comparing Model Results

There are numerous combinations of fields and field similarities available for use with both individuals and households, shown here:

Fields	Individual Similarities	Household Similarities
Relation to head of household	Jaro-Winkler	Jaccard
Religion	Exact Match	Ruzicka Jaccard
Education	Soundex	Group Linkage
Age	TF-IDF	
Gender		
Occupation		
Birthplace		
Marriage status (1911 only)		
Children born (1911 only)		
Children alive (1911 only)		

We do not perform model selection to try to determine which subsets are best and instead focus on comparing models, given a constant set of fields / similarities that are both common in the literature and work well in this application. As we mentioned earlier in the section, we will use fields that we found (in [21]) that have little missing data. The sets of variables that we use in susequent analyses are listed below.

Individual Variables: Forename Jaro-Winkler, Surname Jaro-Winkler, Age Adjusted-Difference
Sex Exact, Birthplace Jaro-Winkler, Location Jaro-Winkler

Household Variables: Forename (Ruzicka) Jaccard, Surname (Ruzicka) Jaccard,
Age (Ruzicka) Jaccard, Sex (Ruzicka) Jaccard,
Birthplace (Ruzicka) Jaccard, Location (Ruzicka) Jaccard

Naive Group Link: Group Linkage (Eq. 5.16), s.t. $sim(a_i, b_j) = avg(sim(\boldsymbol{X}_i^{1901}, \boldsymbol{X}_j^{1911}))^*$

Group Link: Group Linkage, s.t. $sim(a_i, b_j) = \Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911})$

The first set of variables, titled "Individual", are individual pairwise similarities. The "Household" variables are household pairwise similarities calculated directly from the original records. The equation for Ruzicka Jaccard can be found in Equation 5.8 and the original Jaccard is defined in Equation 5.7. We can also add the "Group Linkage" metric as a feature in our classification models. "Naive Group Link" uses individual pair similarity scores to calculate the house similarities, and "Group Link" uses individual pair match probabilities to calculate it. In cases where we calculate "Group Link" from a statistical model, we split the *training* data into a 20%/80% split where the first 20% is used to calculate a baseline individual model. From this baseline model we can calculate the predicted probabilities for the remaining training data and then calculate the group linkage score for each household pair. This remaining 80% of the training data is used to build a model with group linkage as a variable.

In Section 4.1.2 we introduced various ways to consolidate labels collected from our interface. In this section, we utilize the "Total Match" calculation from Table 4.6. This is defined as follows:

$$\text{Total Match Proportion}(i,j) = \frac{\#\text{ of Page 1 Matches}(i,j) \ + \ \#\text{ of Page 2 Matches}(i,j)}{\#\text{ of Page 1 Labels}(i,j) \ + \ \#\text{ of Page 2 Labels}(i,j)}$$

$$\text{Total Match } = \ y_{ij} = \begin{cases} 0 & \text{if Total Match Proportion} < 0.5 \\ 1 & \text{if Total Match Proportion} \geq 0.5 \end{cases}.$$

The pair of individuals ($\boldsymbol{X}_i^{1901}$, $\boldsymbol{X}_j^{1911}$) is labeled as a $y_{ij} = 1$ if the the proportion of matching labels across both Page 1 and Page 2 is greater than or equal to 0.5 and a $y_{ij} = 0$ if it is less than 0.5. We take this approach as a first step because when label quality is unknown, this is the most common approach. Since we do have information on label quality, We explore other definitions in Chapter 7.

6.5.1 Comparing Model Performance

We first compare the classification models of logistic regression, hierarchical models, and random forests. We compare these models using two sets of variables: one without and one with household-pair comparisons. The sets of variables are listed below and are referred to as "Individual" and "Household" within the resulting figures.

In Figure 6.1 we explore the recall, precision, and AUC for logistic regression and hierarchical models, both with and without household-pair variables. The logistic regression model without household variables had an AUC of 0.6 but that slightly increases to 0.623 with the addition of household information about the individual pairs we model. The AUC for the hierarchical model with fixed effects for only individual variables is 0.7609 but by including household variables as fixed effects, this drops to 0.7511. In both hierarchical models, the household-pair ID is used as the random effect. Any household pair in the testing data that did not appear in the training was predicted using only fixed effects. We find that the hierarchical model outperforms the logistic regression model, but including household information as fixed effects does not improve performance.

We can compare these results to those of a random forest model, shown in Figure 6.2. We find that the random forest with only individual fields has an AUC of 0.661 but this increases to 0.744 and 0.754 with

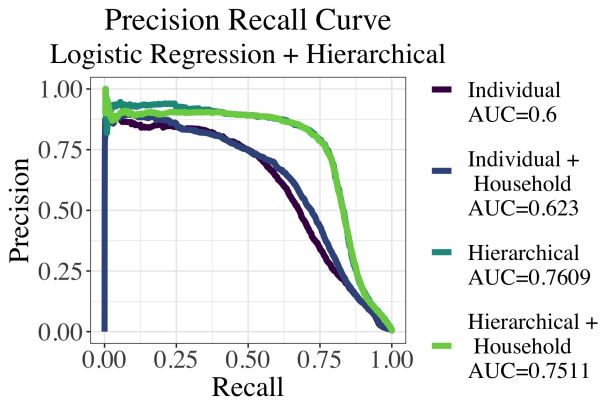


Figure 6.1: AUC, precision-recall graph for logistic regression and multi-level models, without any additional blocked pairs. Household covariates are helpful to the logistic regression model, but are not necessary in the hierarchical model (as fixed effects). Overall, the hierarchical models perform better than logistic regression.

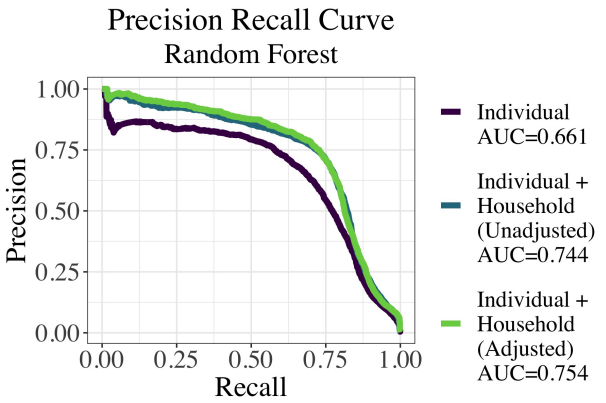


Figure 6.2: AUC, precision-recall graph for random forest models, without any additional blocked pairs. We add adjusted and unadjusted Jaccard household covariates. Both make improvements upon the model without any household information.

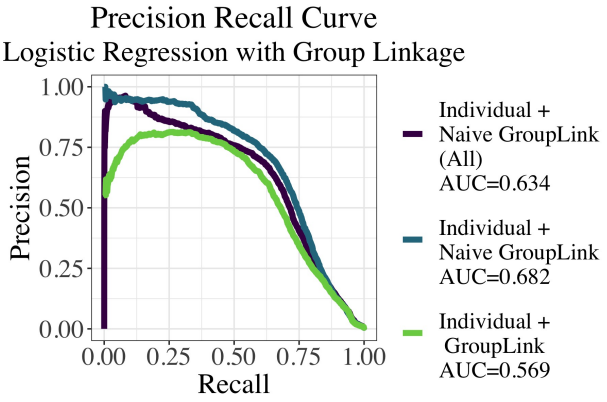


Figure 6.3: AUC, precision-recall graph for logistic regression models, without any additional blocked pairs. We add household information via the group linkage metric. We both naively calculate group linkage and calculate it using a preliminary statistical model (using a subset of the training data). In the case of "(All)" the group linkage for the training and testing were calculated together. This is important to note because households are often split across training and testing.

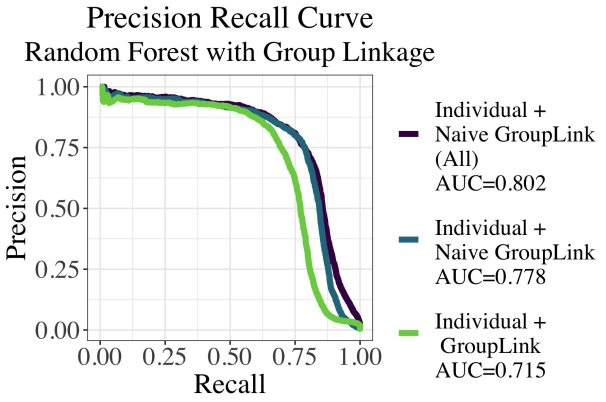


Figure 6.4: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Naive group linkage outperforms model-based group linkage.

the addition of household similarity fields. We do not detect a large difference between results using the unadjusted and adjusted Jaccard similairites. As mentioned in Section 5.3.3 it is more practical in record

linkage, and specifically in this application, to use the adjusted comparisons so we proceed with the adjusted Ruzicka Jaccard similarities.

We also explore model performance for logistic regression and random forest models with the addition of a group linkage field. Adding in group linkage is a way to account for household similarity by utilizing information about the similarity of pairs across households. We use the phrase "(All)" in the legend to note that the group linkage was calculated for the training and testing sets together. This means that households were subset and compared before training and testing splits. This is the same way that we calculate Jaccard similarities. But, in the case without "(All)" we calculate the group linkage of a household pair only for individual pairs from the house that were in the respective training or testing groups. The fact that the Naive Group Link that where the scores were calculated separately in training and testing perform better than the case where group linkage is calculated with everyone is surprising. I would have guessed that the group linkage metric would better represent the household similarity in cases where it's calculated with all members. We also find, in Figure 6.3, that the Naive group linkage model performs better than model that calculated group linkage from a statistical model. Similarly for random forests in Figure 6.4, we find that the Naive group linkage performs better than the statistical group linkage. However, in the random forest models, the model where group linkage was calculated with the training/testing data together performs better than the model where they were calculated separately.

Figures 6.1, 6.2, 6.3, 6.4 all show results for the 40% holdout testing data that was described in Section 6.4. Tables of variable importance and regression coefficients for the linear models can be found in Appendix Section B.1.

In Section 4.2 we introduce blocking to the comparison space. For reference records with very common names, up to almost $8,000$ rows of data could be added, which we saw in Figure 4.1. Because of this imbalance we also wanted to examine results with only a random selection of blocked pairs added for each reference record. Therefore, we randomly sample up to 10 and 25 of the additional blocked pairs (per reference record) and analyze the effect of models with only those pairs added too. Note that the previous models were built without these additional rows added for blocking. The additional rows from blocking are labeled as $y = 0$ because we have determined them to be non-matches. We present the results for a random forest with blocking in Figures 6.5 and 6.6. Within the legend you will find different colors for "Block 10", "Block 25", and "Block All", representing the two subsets and the full blocking pass.

We notice that the AUC is lowest in the cases where all blocked individuals were added to the model. The models perform similarly in the cases where only 10 or 25 blocked pairs were added. In Figures 6.7 and 6.8 we see similar trends for logistic regression and hierarchical models. In a few of the models the 10 block case is slightly better than the 25. Overall, these results make sense; furthering the class imbalance by

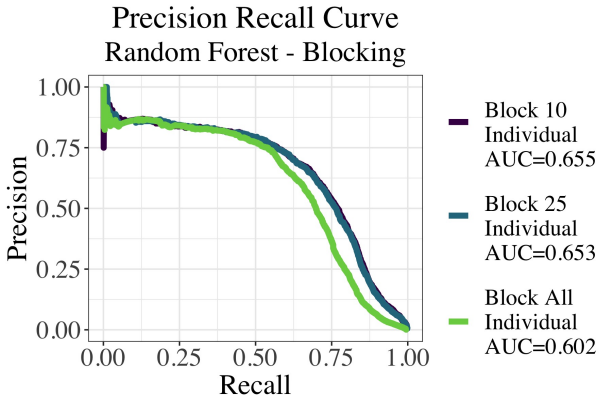


Figure 6.5: AUC, precision-recall graph for random forest models, with additional blocked pairs. Models did not include any household-related covariates.

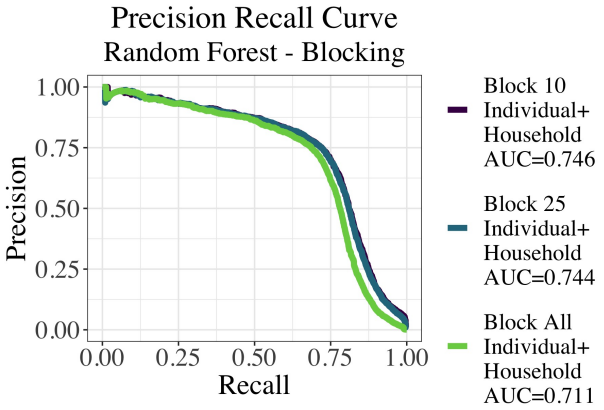


Figure 6.6: AUC, precision-recall graph for random forest models, with additional blocked pairs. Models include household-level adjusted Jaccard covariates.

adding more zeros to our training data only makes for, on average, worse models. These differences are less apparent between cases where only 10 or 25 blocked pairs per reference record are added. Since blocking is often applied as a necessary step, we argue that the strictness of blocking plays a critical role in downstream model outcomes and should be paid attention to.

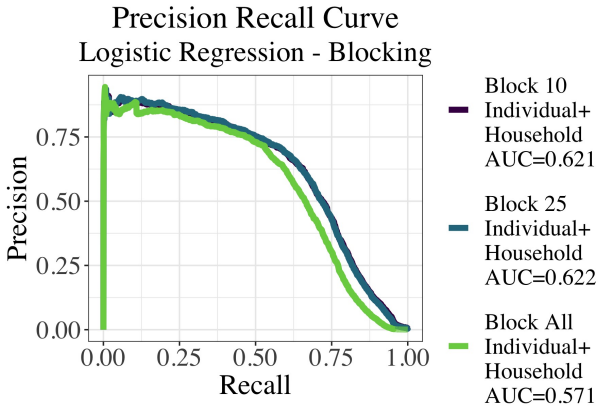


Figure 6.7: AUC, precision-recall graph for logistic regression models, with additional blocked pairs. Models include household-level adjusted Jaccard covariates.

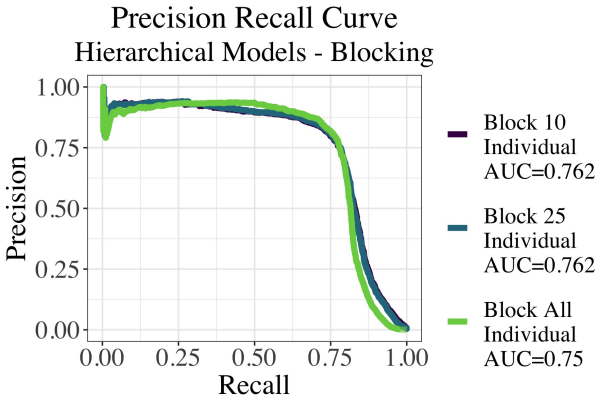


Figure 6.8: AUC, precision-recall graph for multi-level hierarchical models, with additional blocked pairs. Models do not include fixed effects for household-level information, but use the household pair ID as a random effect.

6.5.2 Model Results for Hard-To-Label Sub-Populations

Certain sub-populations are known within record linkage to be especially hard to label correctly. These populations include individuals who move between census years, women who change their last names after marriage, and individuals who work in households different than their home residence (e.g., servants). In Figures 6.9 and 6.10 we explore how models perform different only individuals who move between census dates. Due to the very large out-of-sample data sizes, we have randomly selected $3,000$ pairs to show you in both graphs[†]. The results for the full data were similar, but the crowded graphs made them harder to read.

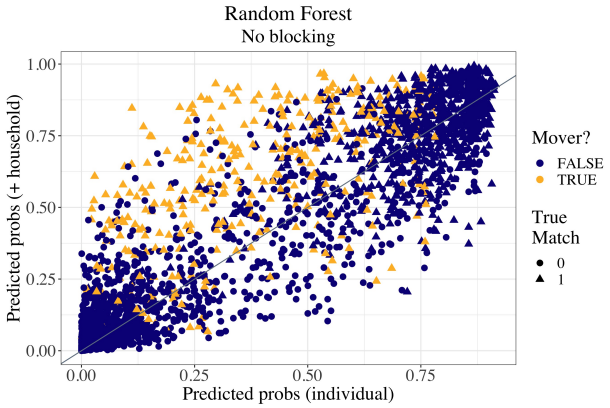


Figure 6.9: Random forest models with (y-axis) and without (x-axis) household information. Models with household information tend to identify "movers" better than models without it. This figure was created via a subset of 3,000 pairs but the same trend is seen on the full holdout data. Additionally, no blocked pairs were added here.

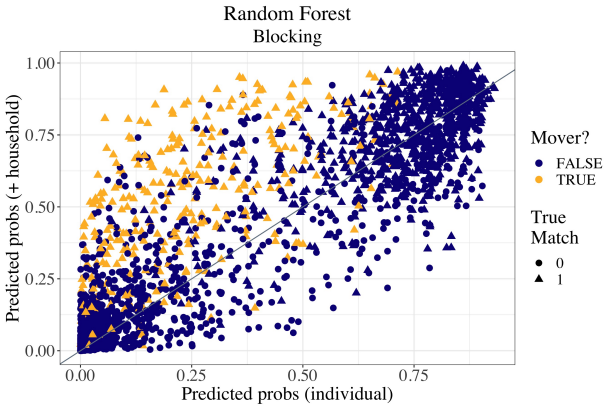


Figure 6.10: Random forest models with (y-axis) and without (x-axis) household information. Models with household information tend to identify "movers" better than models without it. This figure was created via a subset of 3,000 pairs but the same trend is seen on the full holdout data. Blocked pairs were added here.

In both graphs the x-axis represents predicted probabilities from the individual (baseline) random forest model and the y-axis shows the predicted probabilities from the model that additionally includes the Ruzicka Jaccard household similarity fields. We define movers as any pair that is correctly linked together, but has different locations between censuses, and color these points in yellow. In both graphs we can see a large cluster of yellow points in the upper left corner meaning that the model with household information correctly identifies movers that the baseline model does not. This is evidence that the model with household information not only performs better in terms of AUC, but it identifies movers at a higher rate.

We are additionally interested in how well our models identify servants. Shown in Figures 6.11 and 6.12 we show servants who were matched together by labelers in dark purple triangles. All other pairs are shown

[†]Samples were selected randomly, with a weight on the predicted probability so that points with higher predicted probability would be selected at a larger proportion than those with a low predicted probability. We did this because there are so many pairs that are predicted to be non-matches.

in lime green. True matches are shown with triangles and true non-matches are shown with circles. In Figure 6.11 we see that the random forest and the random forest with household information both perform similarly at identifying servants. There are servants that both models identify and servants that both models miss. But, in Figure 6.12 we compare the random forest with household information to the hierarchical model that uses household as a random effect. We see that there is a small cluster of servants that are only identified by the hierarchical model.

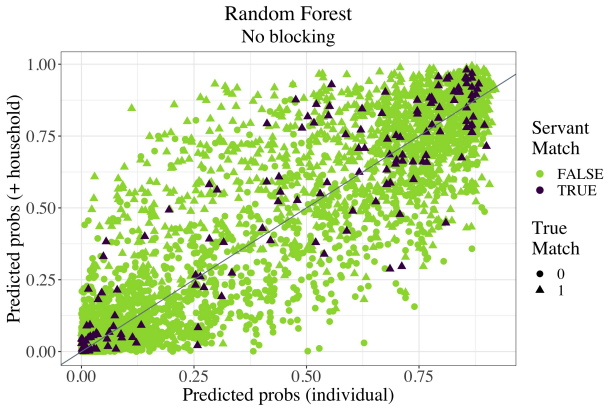


Figure 6.11: Identifying servants across random forest models with (y-axis) and without (x-axis) household information.

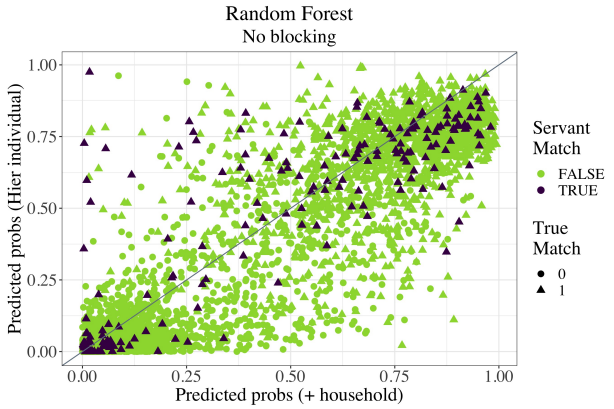


Figure 6.12: Identifying servants across random forest models with household information (x-axis) and hierarchical models with household pair as a random effect (y-axis). In the upper left corner we observe that the hierarchical model identifies some servants that the random forest does not.

6.5.3 Comparing Predicted Probabilities of Models

In Figures 6.2 and 6.1 we found that random forests with household-level covariates performed similar to the model of the hierarchical model. Adding naive group linkage as a covariate improved performance even more. All models performed better than baseline models without household information. Given that all three approaches included household information, we explore the differences between models in Figures 6.13 and 6.14. Each point represents an individual pair and the x-axis and y-axis are the predicted probabilities from the various models. Points are closed circles for non-matches and triangles for matches.

While there is an overall linear relationship between the model outputs, the models perform differently for some pairs. In Figure 6.15, there is a cluster of points in the upper left hand that the random forest does

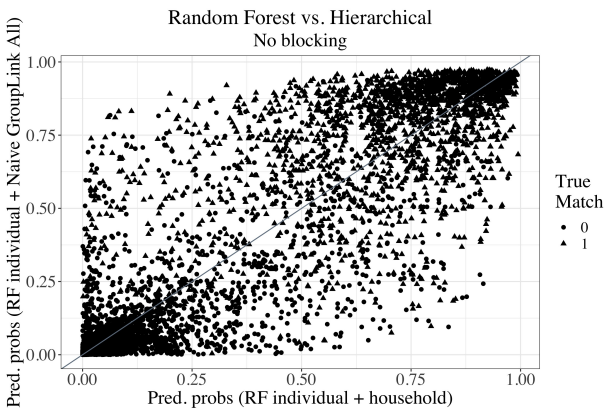


Figure 6.13: Comparing predicted probabilities of a random forest with adjusted jaccard household-level covariates to a random forest with naive group linkage household-level covariates.

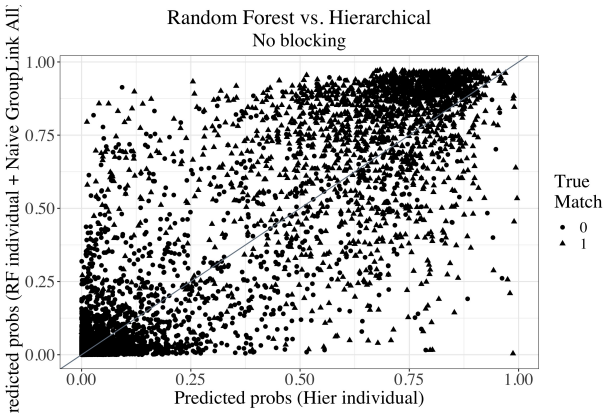


Figure 6.14: Comparing predicted probabilities of a hierarchical model with random effects for household pairs to a random forest with naive group linkage household-level covariates.

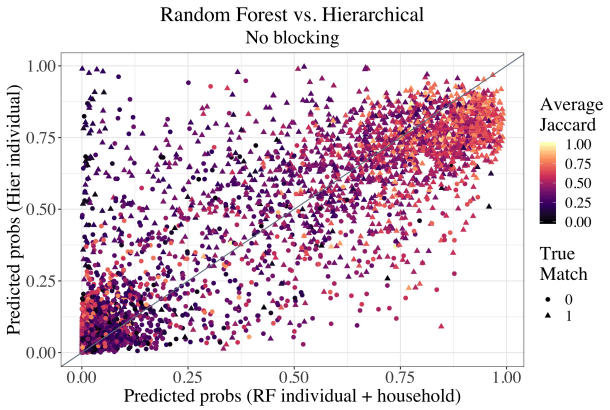


Figure 6.15: As we saw in Figure 6.12 the hierarchical model and random forest with household information performed differently in terms of predicted probabilities and ability to identify servants. We examine here the average adjusted Jaccard similarities across the pairs.

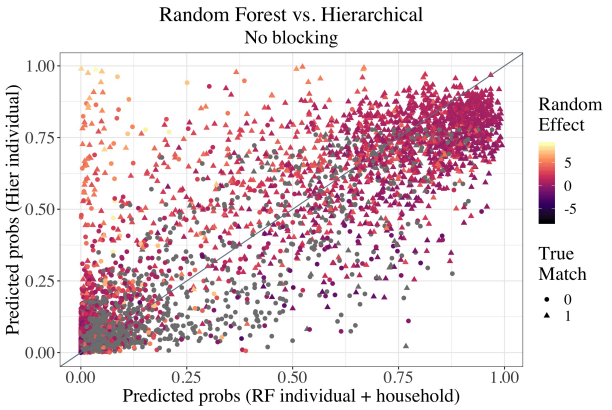


Figure 6.16: We can also examine the random effects from any pair that had household members in the training data.

not believe should match, that the hierarchical model is confident should match. Upon further exploration, many of these pairs were of individuals who were servants although there we were unable to determine a clear reason why the multilevel model matched those individuals but the addition of Jaccard similarity did not.

A common followup question might be whether an ensemble method combining the two models would perform better than either method alone alone. Simply taking the average of the two predictions, wee see that the AUC is higher for this ensemble in Figure 6.17. We could also assign weights to the two models depending features of the individuals (e.g., servants get a higher weight for hierarchical).

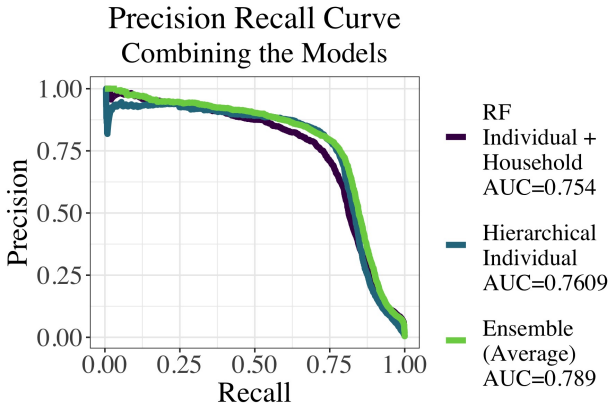


Figure 6.17: AUC, precision-recall graph for a model that averages the predicted probabilities of the random forest with household Jaccards and the hierarchical model.

6.5.4 Train Test Split by 1901 Household

As we mentioned in Section 6.4 one of the biggest issues with splitting the data by reference record is that households can get split across training and testing. That is, in a family of four the parents' comparisons could end up in the training and the children's comparisons could end up in the testing. Therefore, we also wanted to explore model performance where we split our training and testing by household. When we do so, there are $984,225$ comparisons in the training set and $734,253$ in the testing giving a similar ratio to our 60%/40% split from before. There are $3,257$ unique 1901 households in the training data and $2,174$ 1901 households in the testing. When we split by reference record there were $4,173$ and $3,406$ households, respectively. A challenge of splitting by household instead of individual is that the distribution of number of pairs per splitting unit increases drastically, as shown in Figures 6.18 and 6.19. The maximum number of pairs per reference record is $3,562$ but per household is $1,751,507$. Therefore we have to be cautious about maintaining an even class balance when splitting this way.

After we've split the data by household such that no 1901 household (i.e., the individuals within it) is in both the training and the testing, we can evaluate results for various models. In Figure 6.20 we see model results for logistic regression models both with and without household information. Neither model performs well, and both perform worse than when we split on reference record (as a reminder, those AUC values were 0.6 and 0.623). When exploring the results for the random forest in Figure 6.21, we see that the baseline model with individual variables also does worse than when we split on reference record. However, the random forest with household information has a much higher AUC (of 0.888) compared to the AUC of 0.754 when we split on reference. Potentially keeping the households together created more cohesive training and testing

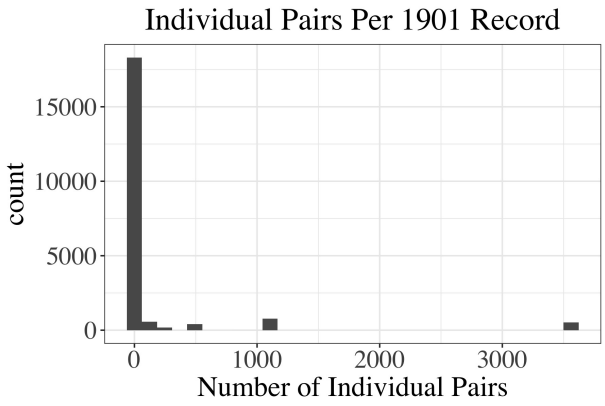


Figure 6.18: The distribution of pairwise comparisons per 1901 reference record. Most records have few comparisons, but some have over 3,500.

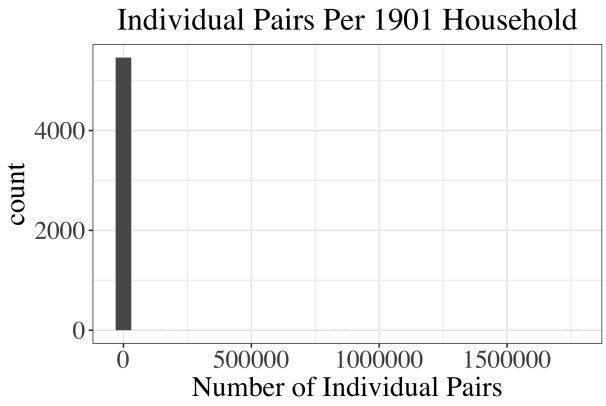


Figure 6.19: The distribution of pairwise comparisons per 1901 household. Most records have few comparisons, but some have over 1.5 million. This greatly skewed distribution makes it difficult to split by household.

sets that looked more similar in distribution. Either way, is important to pay attention to training / testing split when reporting results.

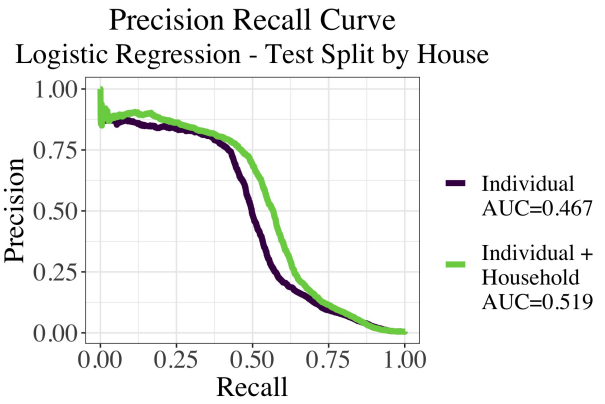


Figure 6.20: AUC, precision-recall graph for logistic regression models, without any additional blocked pairs. We explore the curves both with and without household Jaccard covariates. We split the training and testing data by household so that individuals within households are not split across the training and testing sets.

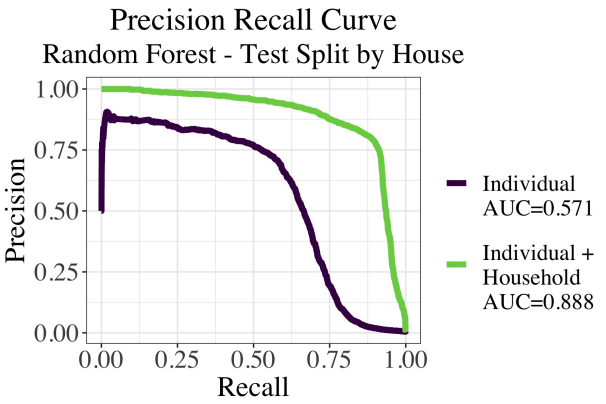


Figure 6.21: AUC, precision-recall graph for random forest models, without any additional blocked pairs. We explore the curves both with and without household Jaccard covariates. We split the training and testing data by household so that individuals within households are not split across the training and testing sets.

We can also see how models with group linkage variables perform when we split by household. In Figure 6.22 we explore model results for random forests with the group linkage variable, calculated both naively as well as from an initial random forest model on 20% of the training data.

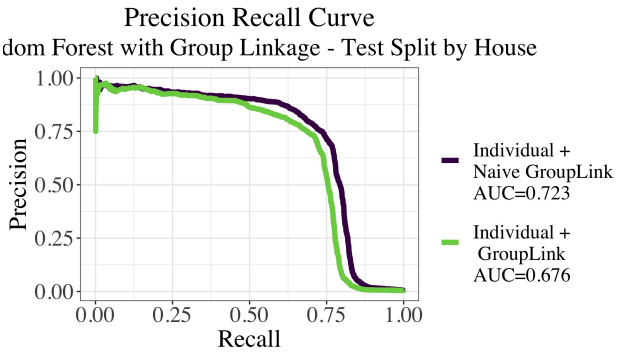


Figure 6.22: AUC, precision-recall graph for random forest models, without any additional blocked pairs. We explore curves for the models with the naive and the model-derived group linkage covariates.

Interestingly, the random forest with Jaccard household similarity appears to do better when we split by household but the random forest with group linkage household similarity performs worse.

6.6 Concluding Thoughts on Modeling

In this chapter we detailed how to pre-process data to be suitable for record linkage models. We introduced how to incorporate household information into statistical classification models by using the Jaccard index and group linkage measure. We also introduced a multilevel model as a third way to include household structure to the modeling process. We discussed model validation and the advantages and disadvantages to various training / testing splits. We then model our data with the classification approaches introduced earlier in the section. We find that random forests outperform logistic regression and that hierarchical models perform similarly to random forests. However, we identify that these models incorporate household information in very different ways leading to differing model predictions on our holdout set. We briefly discuss model performance when we split our training and holdout data by household as opposed to reference record and find that the random forests with Jaccard similarity coefficients for the households perform better, but other models perform worse.

Chapter 7

Interface Analysis

In this section we explore performance of record linkage models based on changes in interface information. We use the same train/test split by reference record that we did at the beginning Section 6.5.1. As we've mentioned, there are advantages and disadvantages to blocking by reference as opposed to household, but we wanted to allow for the possible use of hierarchical models in this section by splitting on reference record. We do not use any blocking in this section, but analyzing this section with the addition of blocked pairs is an interesting area of future research.

7.1 Label Quality

We collected labels over multiple rounds from 2018-2020 and we found evidence of differences in label quality by round (Sections 3.2, 4.3). Early rounds of labels collected via Amazon MTurk had more "bad" or low quality work that often did not pass our approval criteria. In Chapter 6 we made the conscious decision to keep all collected data to understand how our models would perform, given that we used a crowdsourcing platform with various work quality. Part of the reason we collected multiple labels per reference record was to assess the extent to which label quality effected downstream models. Unfortunately we were not able to collect multiple labels for every reference / candidate pair. Shown in Figure 7.1 we see that most pairs were only labeled by one person but that many were labeled by two labelers. The maximum number of labels for a pair of records was 21 labels. On the right in Figure 7.2 we explore the percentage of submits that were from workers who, at one point, did not pass our quality checks. Note that approval criteria was developed through an iterative analysis of label information, so therefore some low quality labels could have been approved in early stages. We also do not argue that our approval classification process identified all forms of poor quality labels. Regardless, between 25 to 40% of labelers in the "MTurk" and "MTurkLong" rounds were from rejected workers. Given the varying label quality across labels, we might be interested in

how often label for a reference record differ. We explore this idea in Figures 7.3 and 7.4. Only looking at labels from Page 1, we see that most often labelers agree but that 11% of the time two unique candidates were given to the same reference record. This drops to 0.25% and 0.01% of the time for three and four distinct labels. Including labels from Page 2, we see that there was one case where six distinct labels were given to a particular reference record. Please note that selecting "no match selected" counts as a unique label in these graphs.

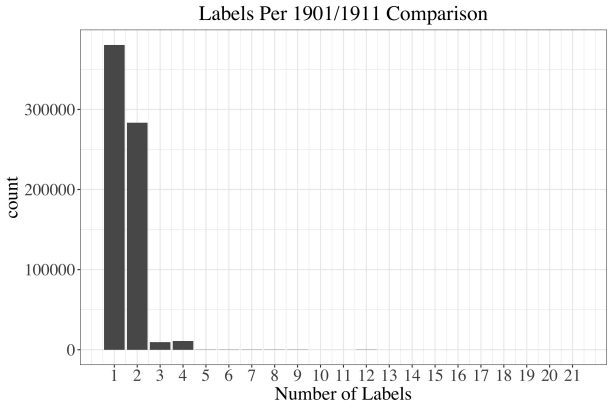


Figure 7.1: Number of labels collected per 1901 / 1911 individual comparison. Most pairs were labeled by only one or two unique labelers but some pairs had up to 21 labelers.

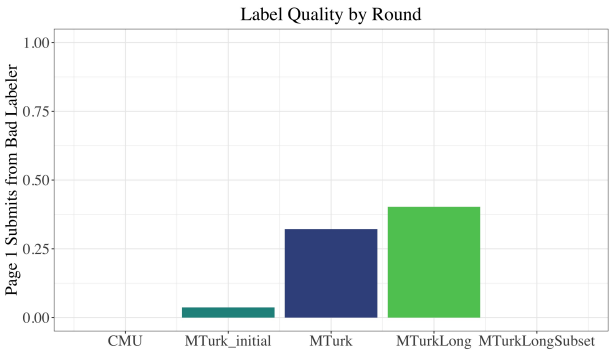


Figure 7.2: Proportion of work per round that was completed by poor quality labelers. In the initial round we approved most work while we identified systematic patterns to label quality.

Since we observed that there is varied quality labels in the data, we explore how having at least one "bad" label appeared in our models. In Figure 7.5 we explore the relationship between model predicted probabilities and bad labels, given our model's mistakes. On the x-axis we have the predicted probabilities for our random forest without household Jaccard variables, and the y-axis is the predicted probabilities for the model with that household information. This graph is subset to only show mistakes that either model made (i.e, true matches classified as non-matches and vice versa). We color the points by whether the pair had at least one bad labeler. We see a cluster of green points, especially on the right of the figure, indicating that bad labels were prevalent among our model's mistakes. On the right, in Figure 7.6 we further subset the data to only show model predictions for bad labelers. We explore how many unique labelers there were for each pair. We find that for many of the mistakes, there was only one bad labeler; this indicates that the bad labelers had a large influence on these mis-labeled points.

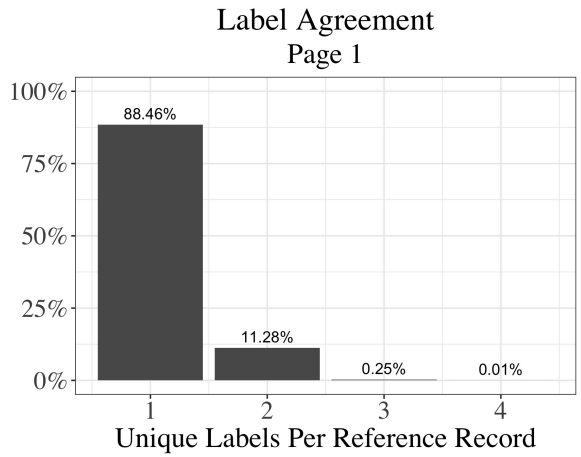


Figure 7.3: Unique labeling decisions per 1901 reference record. A value of "1" indicates that all labelers chose the same candidate (or lack of candidate) for this reference record. Results are shown for labels from Page 1 only.

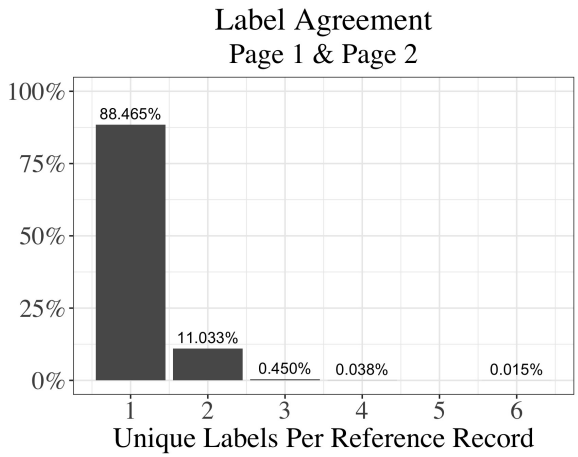


Figure 7.4: Unique labeling decisions per 1901 reference record. A value of "2" indicates that among all labelers, two unique decisions were made about the correct match. Results are shown for labels from both Page 1 and Page 2.

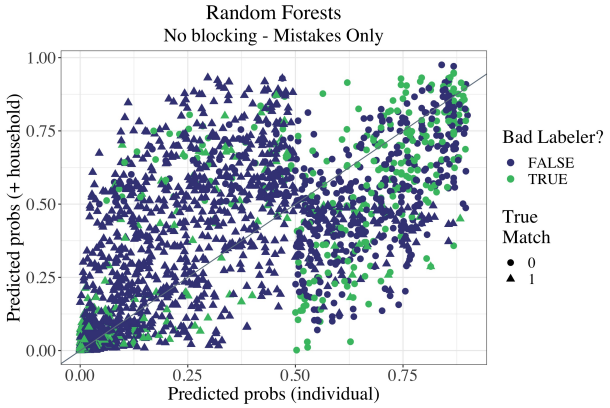


Figure 7.5: Exploring the predicted probabilities for ranadom forest models with and without household information. We subsetted the data to only include mistakes. Pairs are colored by whether their "correct" label was determined by at least one low quality label.

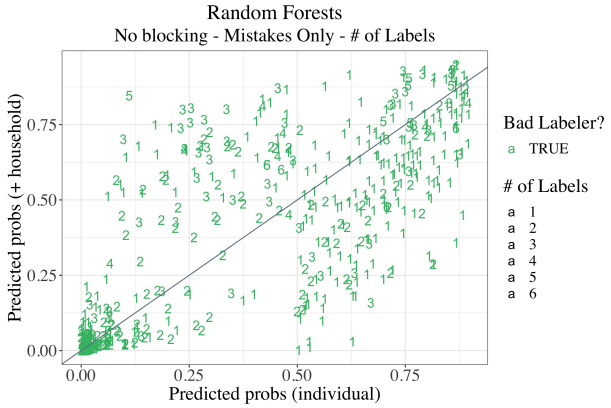


Figure 7.6: Further subsetting to include only the pairs that had at least one low quality label, we see that many of these points had only one labeler. The text on the graph shows the total number of labelers for each pair. These points were therefore unable to be outweighed by higher quality labels.

Given this information, we built a model where we recalculated "Total Match" without any of the labelers who failed our quality checks. The results of this model are shown in Figure 7.8. We see an improvement in AUC compared to our results from the full set (shown again in Figure 7.7). But, we want to understand if this model is better or purely the training data was easier to predict on. In Figures 7.9 and 7.10 we explore the AUC when we predict on the recalculated testing data and the original testing data for the two models. Interestingly, the original model performs just as well as the updated model on the recalculated testing data,

but the updated model performs worse on the full test set. This provides evidence that having bad labels in the training data helps when predicting on data that has bad labels.

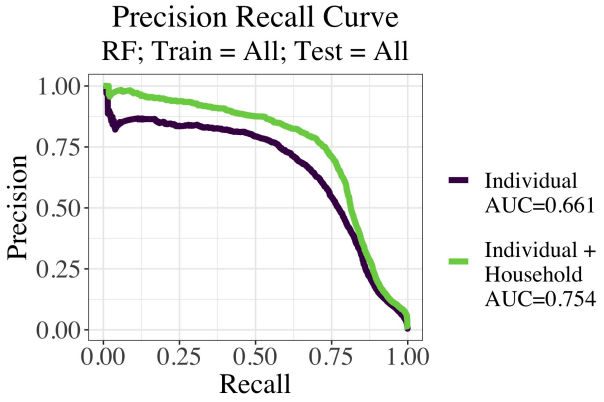


Figure 7.7: Repeating Figure 6.2 from Section 6.5.1 for ease of comparison.

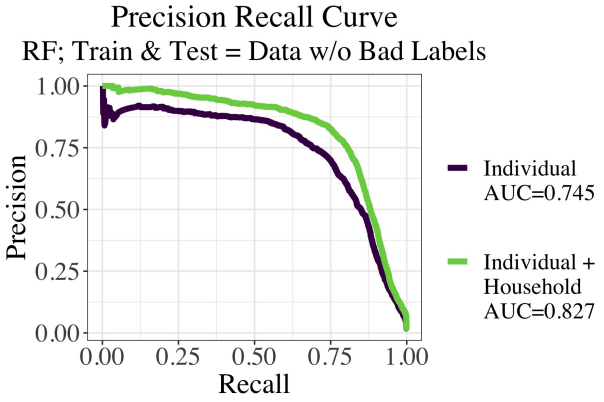


Figure 7.8: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Data was modeled on the subsetted training data where matches were determined without low quality labels. We explore results both with and without household Jaccard covariates for testing data that also had bad labels removed.

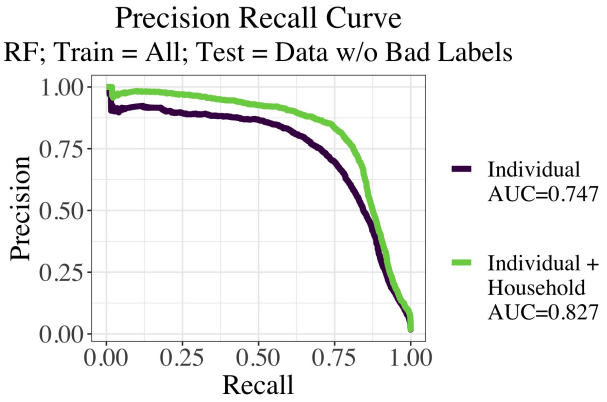


Figure 7.9: AUC, precision-recall graph for random forest models, without any additional blocked pairs. This is the same model as in Figure 7.7, but we remove bad labels from the testing data.

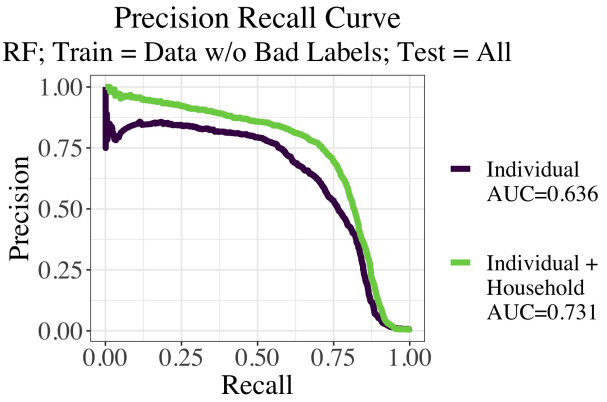


Figure 7.10: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Data was modeled on the subsetted training data where matches were determined without low quality labels. We predict on the full holdout data set that included bad labels.

7.2 Label Source and Round

Using our application, labels are collected on both Page 1 and Page 2 of the interface. These labels come from very different distributions, provided that Page 2 labels are only collected after a household is matched on Page 1. We model the data separately for pairs that were collected on Page 1 and pairs that were collected on Page 2. The $y_{ij} = 0/1$ cutoff is determined only by Page 1 or Page 2 labels depending on the respective model. The AUC for the Page 1 model (both with/without household Jaccard information) is shown in Figure 7.11 and the AUC for Page 2 is shown in Figure 7.12. While it appears that the Page 2 model performs very well, when we take that model but predict on the full holdout data (see Figure 7.13) that includes y calculated from both Page 1 and Page 2, it performs much weaker.

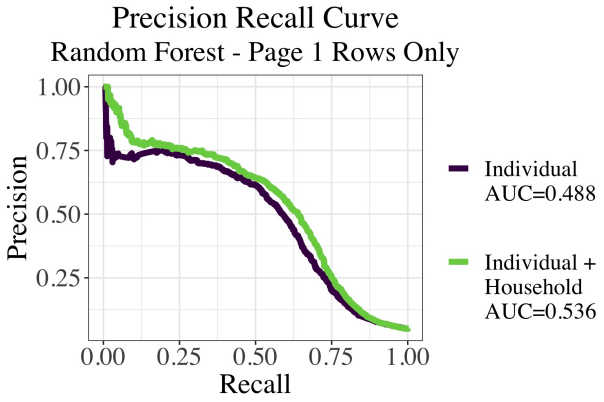


Figure 7.11: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Data was modeled on the subsetted training data including only labels from Page 1. Testing was also from this subset.

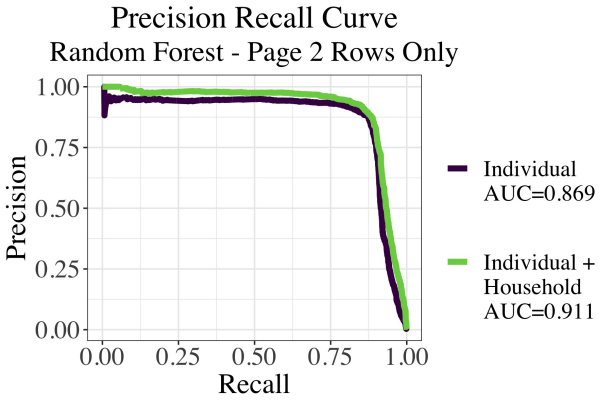


Figure 7.12: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Data was modeled on the subsetted training data including only labels from Page 2. Testing was also from this subset.

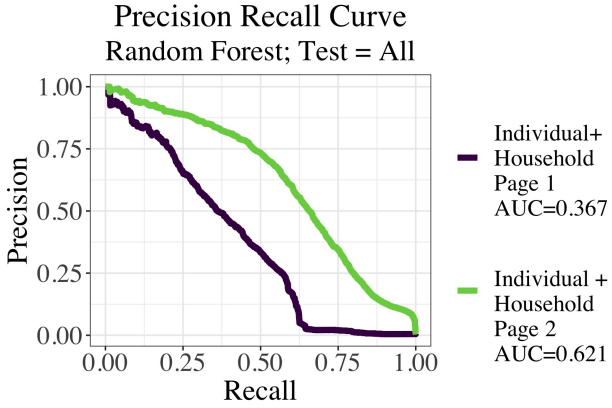


Figure 7.13: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Data was modeled on the subsetted training data including only labels from Page 1 or Page 2 but was tested on the full holdout data.

We were similarly interested in examining labeling differences between CMU and MTurk labelers. In Figures 7.14 and 7.15 we explore the distribution of unique candidates selected per reference record for both CMU and MTurk.

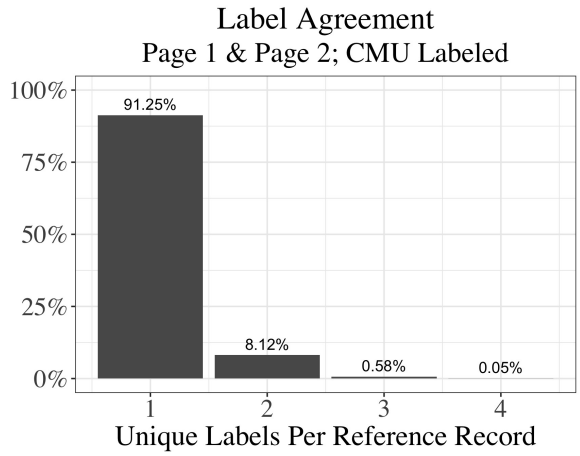


Figure 7.14: Label agreement among CMU labelers.

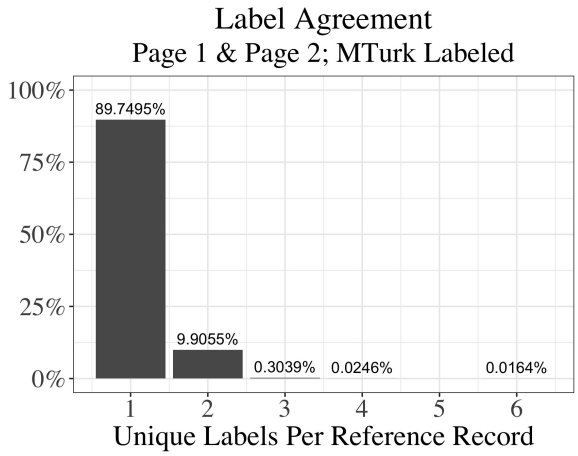


Figure 7.15: Label agreement among MTurk labelers.

In Figures 7.16, 7.17, and 7.18 we explore the AUC, precision, and recall for models using only CMU data and models using only MTurk data. We recalculated the match proportions and true match using these labels only.* The last graph shows labels/data that came only from our best MTurk labelers. We find that models using CMU and the MTurkLongSubset data were better able to recover matching and non-matching pairs. The overall MTurk data struggled in terms of AUC, precision, and recall. This makes sense given that we know many of their labels were rejected for poor work quality.

7.3 Section Conclusion and Modeling Extensions

Recall the baseline model without household information (Section 6.3.1, Figures 6.2 6.1). Next steps might be to incorporate familial network structure or covariates for labeler quality, source, and round into modeling.

*We also added label round as a feature to our random forest no blocking model. While the overall AUC performance slightly improved none of the rounds had variable importance higher than other variables.

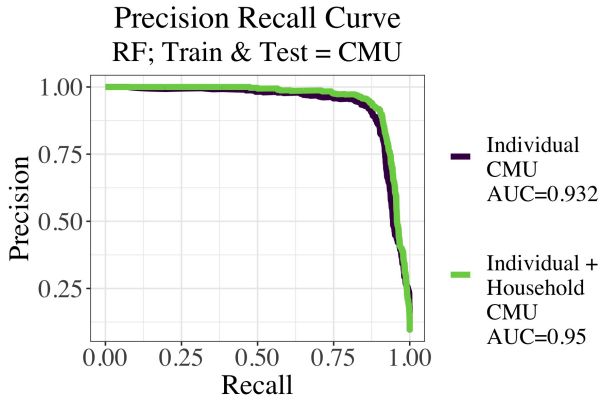


Figure 7.16: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Data was modeled on the subsetted training data including only labels that originated at CMU.

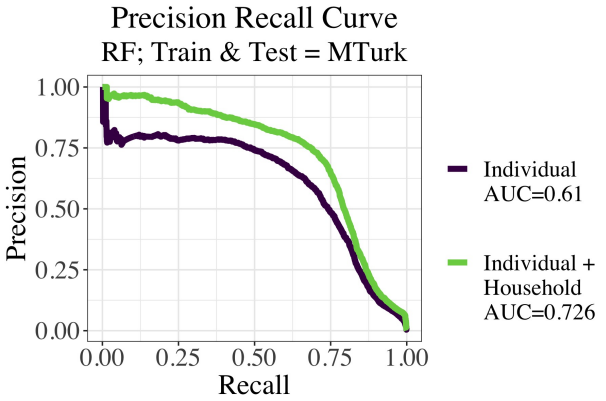


Figure 7.17: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Data was modeled on the subsetted training data including only labels that originated in Amazon's MTurk.

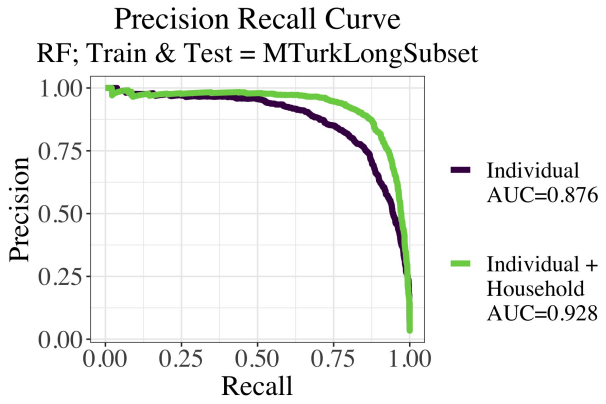


Figure 7.18: AUC, precision-recall graph for random forest models, without any additional blocked pairs. Data was modeled on the subsetted training data including only labels that originated in Amazon's MTurk, but only for the last batch which was our best labelers.

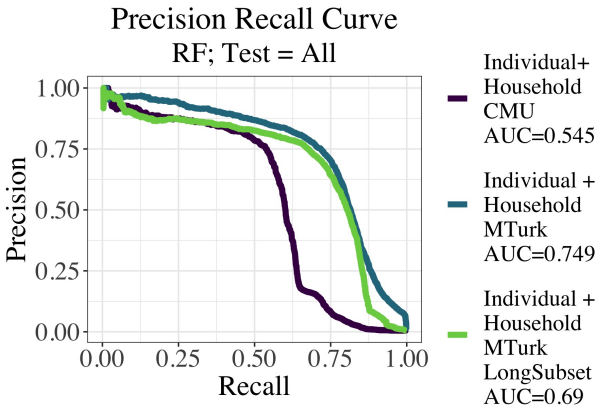


Figure 7.19: Predicting on the full holdout data.

We could model the count of labels instead of the discretized label proportions. We incorporated household information as covariates using both Jaccard similarities and group linkage similarities in Sections 6.3.2 and 6.3.3 and results were shown in Figures B.3 and 6.4. Both of the individual and household covariates could vary by interface information and potentially including this information directly into those models would be beneficial. For example, potentially we could down weight labels that were determined hastily. Or perhaps we do not use household information in the model if we noticed that the labeler never viewed the households. We used multilevel models as a way to incorporate household information in Section 6.3.4. We

could additionally use multilevel models to include a labeler effect, but there are often multiple labelers per pair so incorporation of this effect requires more thought. Additionally, using label generation information as well as the fact that there should only be one 1911 record for each 1901 record, we could extend our methods to enforce a one-to-one match across the censuses and/or deduplicate the individual censuses. In Section 4.1.2 we address the challenges of consolidating labels. In Chapter 6 we proceed to use an average across all labels, but when we combine labels from Page 1 and Page 2, we could attempt to incorporate the fact that Page 2 labels are drawn from a different distribution than Page 1 labels (i.e. $Pr($ individual match | blocking) vs. $Pr($ individual match | household match, blocking)). The answers to many of these questions are non-trivial and require additional thought, but they are interesting extensions of current work.

In summary, in this section we showed that record linkage model performance (AUC, precision, and recall) vary drastically depending on the data you feed in. Data from high quality labels/labelers and Page 2 of the interface all produce relatively well performing models (when we evaluate on data from the same populations). But, if we expect testing data to include poor quality labels, models that were built using poor quality labels perform better. None of these results are surprising but confirm how important labeling decisions are to downstream model performance, despite how often they are (often unavoidably) overlooked.

Chapter 8

Conclusion

8.1 Public Data Access

Information on how to access the data will be added after the dissertation defense.

8.2 Dissertation Summary and Contributions

In this thesis we explore approaches for linking the 1901 and 1911 Irish Census data bases. This data was only released in recent years and therefore there are no existing identifiers to link the records. After attempting to solve this problem in an unsupervised fashion, we realized that collecting labels would be critical to building record linkage models that could accurately separate matching records from non-matching records[22]. Therefore we developed a record linkage interface / platform that can be used to crowdsource labels for this census data. After having spent time hand labeling records for my ADA with the United States Census Bureau, I know that household information is critical to the identification of matching individual pairs. I also learned that matching labels pair-by-pair was ~~mind-numbing~~ time consuming and matching groups of pairs at one time would be more efficient. When we started developing the interface, which was done completely in `R Shiny`, our goals were to collect many pairs of labels at once and to allow labelers to easily incorporate household information into their decision making process. The first round of labels were collected by CMU students and professors and those preliminary analyses can be found in our IEEE ICDM paper [22]. We found evidence that both including household information and the use of our interface are beneficial to the record linkage pipeline. From just the first batch of labelers we saw how much uncertainty was involved in the record linkage process and how seldom this uncertainty was captured. Pre-labeled record linkage data are often taken as ground truth and we found that little is known about the record linkage labeling process. Therefore, the concluding next steps from this first round of labeling were that 1) we

should try to study how our labelers interact with the labeling interface to better understand the entire labeling process, and 2) that we should also collect household matches and individual-within-household matches to provide richer label information at a relatively low cost*.

As we moved into the second phase of label collection, we incorporated the interface changes mentioned above and updated our thesis goals. The second phase of labeling, also conducted by CMU students, led to richer data and the ability to analyze labeler interactions with the interface. We found more evidence of the importance of household information and correctly identified individuals who move locations between census dates at a higher rate than baseline record linkage models. We introduced this second phase of the interface and it's benefits in our IEEE DSAA paper[21].

Although we greatly appreciated the help of our CMU colleagues / labelers, we knew that more (albeit delicious) Indian food would not be enough to encourage a third round of labeling. Additionally, we were interested in studying how labelers in less controlled environments reacted to our linkage interface. Therefore, we started utilizing Amazon's Mechanical Turk platform to source online work from around the world. This provided a fast way to collect a large amount of data, for a *relatively* reasonable price. Utilizing this interface came with challenges (which we mention throughout Chapter 3), and we had to balance label quality with worker satisfaction. We discussed how we consolidate the multiple sources of data we collectd in Chapter 4, while also reporting the raw data to allow other researchers to consolidate in different ways. In Chapter 5 we discuss how to take pairs of raw data and make mathematical comparisons between them. We do this at both the individual and group level, and make comparisons between the group metrics. In Chapter 6 we introduce the various models we end up comparing, utilizing household information in various ways across the models. We report model results for those models and find that incorporating household information (in any form) greatly improves model results. We discuss how the addition of blocking impacts models. We pay special attention to subgroups like movers and servants. In Chapter 7 we go further to discuss differences in performance based on interface aspects. We explore agreement between labelers, the impact of poor quality labels, and how the data was generated.

This thesis created a novel data set for use within the record linkage community. This data is novel in multiple ways. Firstly, we provide nested labels for individual pairs, household pairs, and individual pairs within households. Secondly, we collect and provide detailed information on the record linkage label collection process. Thirdly, we collect label certainty by collecting multiple labels per pair as well as providing quality information about the labelers themselves. We showed how important both household information and the labeling process is to the downstream linkage results and argue that these details should be given greater attention.

*It takes very little time to label households and individuals within households once you have already examined all of the individual records across both households.

8.3 Interesting Areas for Future Work

As with any project, there are numerous interesting extensions. Specifically, because this data provides information that is seldom available in linked record linkage data, there are many interesting areas for future study. We group our thoughts by category below.

Record Linkage Methodology We are very interested in directly incorporating information from the label generation process into record linkage methodology. We introduce multiple proposed extensions in Section 7.3. Additionally we are interested in whether we could/should model the sequence of clicked candidates to identify the most likely ones? What meta information can we learn from these click sequences? We are aware we introduce biases when we label via our interface, but potential extensions include addressing these biases to account for them later on. Can we sequentially link households and then individuals, and impute unknown structure for unseen households? In addition, it would be interesting to incorporate expected field changes (last name changes, household moves) to account for the temporal aspect of this problem. Can we learn these transitions from the data? What is the extent of needing de-duplication within the 1901 and 1911 Irish Census records? How can household / additional structure be used within de-duplication problems? Is it similar to how it can be used across years? In terms of unsupervised linkage, would extending our household similarity metrics beyond exact matching as well as using non-binary comparisons in our E-M estimation help unsupervised models find signal in this data?

Household Similarity Metrics What is the consequence of using a stringent cutoff for matching pairs in group linkage? Can we incorporate information from non-matches while weighting matches higher? What information about the non-matches would be helpful in making a decision about two households and would we even want to include this? Does this stringent cutoff effect large and small households differently? We identified that classification models use household information (e.g., Jaccard, Group Linkage) in different ways, but *how* do they differ and can we capitalize on their unique differences? We found that the naive group linkage performed better than group linkage similarities derived from statistical models; can we better understand this relationship?

Tuning the Labeling Process One potential next step of future labeling with this interface is the incorporation of active learning into the label selection process to attempt to collect labels that help downstream modeling. What do informative matches and non-matches look like in historical record linkage? If we were to continue labeling via Amazon Mechanical Turk, we would hope to identify and utilize strong labelers early in the process to increase the quality to cost ratio. We provide rich information on labeler and time sequences and could study the time on task fatigue as an alternative way to quantify label quality. It would be interesting to explore patterns by labeler. Please note that IRB approval was not needed for our work, but depending on future label experiments / studies it might be needed.

In conclusion, it is important to me that this data is made public. It is my top priority to make this data accessible and promote its use across various areas. I am excited for future extensions and collaborations and am beyond grateful for the support of my committee through this process.

Bibliography

[1] (2005).

[2] (2007 (Retrieved 2010-12-24)). The soundex indexing system.

[3] (2008). r/mturk.

[4] (2011). UCI machine learning repository; record linkage comparison patterns data set.

[5] Abramitzky, R., Boustan, L. P., and Eriksson, K. (2014). A nation of immigrants: Assimilation and economic outcomes in the age of mass migration. *Journal of Political Economy*, 122(3):467–506.

[6] Agresti, A. (2003). *Categorical data analysis*, volume 482. John Wiley & Sons.

[7] Antonie, L., Gadgil, H., Grewal, G., and Inwood, K. (2016). Historical data integration a study of wwi canadian soldiers. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 186–193, Barcelona, Spain. IEEE.

[8] Antonie, L., Inwood, K., Lizotte, D. J., and Ross, J. A. (2014). Tracking people over time in 19th century canada for longitudinal analysis. *Machine learning*, 95(1):129–146.

[] Bhattacharya, I. and Getoor, L. (2004). Iterative record linkage for cleaning and integration. In *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '04, page 11–18, New York, NY, USA. Association for Computing Machinery.

[9] Bilenko, M., Kamath, B., and Mooney, R. J. (2006). Adaptive blocking: Learning to scale up record linkage. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 87–96.

[10] Borg, A. and Sariyar, M. (2019). *RecordLinkage: Record Linkage in R*. R package version 0.4-11.2.

[11] Casler, K., Bickel, L., and Hackett, E. (2013). Separate but equal? a comparison of participants and data gathered via amazon's mturk, social media, and face-to-face behavioral testing. *Computers in Human Behavior*, 29(6):2156 – 2160.

[12] Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions.

[13] Christen, P. (2012). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media.

[14] Christen, P. (2012). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Transactions on Knowledge and Data Engineering*, 24(9):1537–1555.

[15] Collins, W. J. and Wanamaker, M. H. (2014). Selection and economic gains in the great migration of african americans: new evidence from linked census data. *American Economic Journal: Applied Economics*, 6(1):220–52.

[16] Davitt, M. (1882). *The Land League proposal: a statement for honest and thoughtful men*, volume 15. Glasgow: Cameron & Ferguson.

[Deza] Deza, E. *Dictionary of distances*. Elsevier, Amsterdam, The Netherlands ;, 1st ed. edition.

[18] Dolatshah, M., Teoh, M., Wang, J., and Pei, J. (2018). Cleaning crowdsourced labels using oracles for statistical classification. *Proc. VLDB Endow.*, 12(4):376–389.

[19] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[20] Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

[21] Frisoli, K., LeRoy, B., and Nugent, R. (2019). A novel record linkage interface that incorporates group structure to rapidly collect richer labels. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 580–589.

[22] Frisoli, K. and Nugent, R. (2018). Exploring the effect of household structure in historical record linkage of early 1900s ireland census records. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 502–509, Singapore. IEEE.

[23] Fu, Z., Christen, P., and Boot, M. (2011). Automatic cleaning and linking of historical census data using household information. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 413–420, Vancouver, BC. IEEE.

[24] Gelman, A. and Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*, volume Analytical methods for social research. Cambridge University Press, New York.

[25] Hand, D. and Christen, P. (2018). A note on using the f-measure for evaluating record linkage algorithms. *Statistics and Computing*, 28(3):539–547.

[26] Hand, D. J. (2009). Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine Learning*, 77(1):103–123.

[27] Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models*, volume 43. CRC press.

[28] Hauser, D. J. and Schwarz, N. (2016). Attentive turkers: Mturk participants perform better on online attention checks than do subject pool participants. *Behavior research methods*, 48(1):400–407.

[29] Hernández, M. A. and Stolfo, S. J. (1995). The merge/purge problem for large databases. *SIGMOD Rec.*, 24(2):127–138.

[30] HO, T. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

[31] Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

[32] Hu, Y., Wang, Q., Vatsalan, D., and Christen, P. (2017). Improving temporal record linkage using regression classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 561–573. Springer.

[33] Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50.

[34] Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420.

[35] Knuth, D. E. (1998). *The Art of Computer Programming*, volume 3. Pearson Education.

[36] Li, B. and Han, L. (2013). Distance weighted cosine similarity measure for text classification. In Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., and Yao, X., editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, pages 611–618, Berlin, Heidelberg. Springer Berlin Heidelberg.

[37] Li, P., Dong, X. L., Guo, S., Maurino, A., and Srivastava, D. (2015). Robust group linkage. In *Proceedings of the 24th International Conference on World Wide Web*, pages 647–657. International World Wide Web Conferences Steering Committee.

[] McVeigh, B. S. and Murray, J. S. (2017). Practical bayesian inference for record linkage.

[38] Michelson, M. and Knoblock, C. A. (2006). Learning blocking schemes for record linkage. In *AAAI*, volume 6, pages 440–445.

[39] Mozafari, B., Sarkar, P., Franklin, M. J., Jordan, M. I., and Madden, S. (2012). Active learning for crowd-sourced databases. *arXiv preprint arXiv:1209.3686*.

[40] of Congress, L. (2019). Irish-catholic immigration to america.

[41] of Ireland, T. N. A. Ireland in the early 20th century.

[42] of Ireland, T. N. A. National archives: Census of ireland 1911.

[43] On, B.-W., Koudas, N., Lee, D., and Srivastava, D. (2007). Group linkage. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 496–505. IEEE.

[44] Pearson, E. S., Gosset, W. S., Plackett, R., and Barnard, G. A. (1990). *'Student', A Statistical Biography of William Sealy Gosset*. Oxford University Press, USA.

[45] Raykar, V. C. and Yu, S. (2012). Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, 13(null):491–518.

[46] Rouse, D. P. (2018). personal communication.

[47] Ruzicka, M. (1958). Application of mathematical-statistical methods in geobotany (synthetic processing of recordings). *Biologia, Bratisl*, 13:647–661.

[48] Sadinle, M. (2017). Bayesian estimation of bipartite matchings for record linkage. *Journal of the American Statistical Association*, 112(518):600–612.

[49] Sammut, C. and Webb, G. I., editors (2010). *TF–IDF*, pages 986–987. Springer US, Boston, MA.

[50] Schubert, A. and Telcs, A. (2014). A note on the jaccardized czekanowski similarity index. *Scientometrics*, 98(2):1397–1399.

[51] Sheng, V. S., Provost, F., and Ipeirotis, P. G. (2008). Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622.

[52] Steorts, R. C., Ventura, S. L., Sadinle, M., and Fienberg, S. E. (2014a). A comparison of blocking methods for record linkage. In Domingo-Ferrer, J., editor, *Privacy in Statistical Databases*, pages 253–268, Cham. Springer International Publishing.

[53] Steorts, R. C., Ventura, S. L., Sadinle, M., and Fienberg, S. E. (2014b). A comparison of blocking methods for record linkage. In *International Conference on Privacy in Statistical Databases*, pages 253–268. Springer.

[54] Tai, X. H. and Eddy, W. F. (2018). A fully automatic method for comparing cartridge case images,. *Journal of Forensic Sciences*, 63(2):440–448.

[55] Tai, X. H., Soska, K., and Christin, N. (2019). Adversarial matching of dark net market vendor accounts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 1871–1880, New York, NY, USA. ACM.

[56] Themstrom, S., Orlov, A., and Handlin, O. (1980). Harvard encyclopedia of american ethnic groups. *Cambridge, MA: Belknap*.

[57] Treeratpituk, P. and Giles, C. L. (2009). Disambiguating authors in academic publications using random forests. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 39–48. ACM.

[58] Tromp, M., Ravelli, A. C., Bonsel, G. J., Hasman, A., and Reitsma, J. B. (2011). Results from simulated data sets: probabilistic record linkage outperforms deterministic record linkage. *Journal of clinical epidemiology*, 64(5):565–572.

[59] Ventura, S. L. and Nugent, R. (2014). Hierarchical linkage clustering with distributions of distances for large-scale record linkage. In Domingo-Ferrer, J., editor, *Privacy in Statistical Databases*, pages 283–298, Cham. Springer International Publishing.

[60] Ventura, S. L., Nugent, R., and Fuchs, E. R. (2015a). Seeing the non-stars: (some) sources of bias in past disambiguation approaches and a new public tool leveraging labeled records. *Research Policy*, 44(9):1672–1701.

[61] Ventura, S. L., Nugent, R., and Fuchs, E. R. (2015b). Seeing the non-stars:(some) sources of bias in past disambiguation approaches and a new public tool leveraging labeled records. *Research Policy*, 44(9):1672–1701.

[62] Walker, S. H. and Duncan, D. B. (1967). Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179.

[63] Wang, W. and Zhou, Z.-H. (2015). Crowdsourcing label quality: a theoretical analysis. *Science China Information Sciences*, 58(11):1–12.

[] Winkler, W. E. (1988). Using the em algorithm for weight computation in the fellegi & sunter model of record linkage. Alexandria, VA. American Statistical Association.

[64] Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *Proceedings of the Section on Survey Research Methods. American Statistical Association*.

Appendix

Appendix A

Intransitive Matches

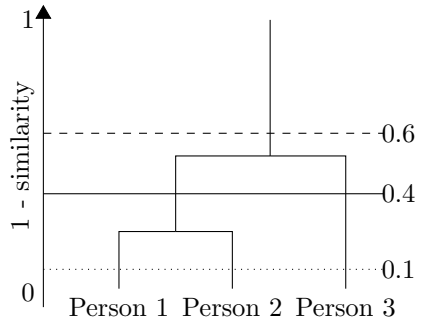
A.1 Resolving Intransitive Matches

Once we have decided whether or not a pair of records "match", or have a high enough similarity/probability of matching, we want to assign unique IDs to the original records. But we need to be wary of the following case:

1901	1911	Similarity
Person 1	Person 2	0.9
Person 2	Person 3	0.6
Person 1	Person 3	0.4

Figure A.1: Intransitive matching problem that occurs with a similarity cutoff of 0.5

Shown in (Fig. A.1), If we use 0.5 for our match probability cutoff, we'd link Person 1 to Person 2, Person 2 to Person 3, but not Person 1 to Person 3. One approach to resolve this transitivity issue, is to hierarchically cluster the records before assigning IDs. This may look like:



Now, we can adjust our cutoff and never have an intransitive match situation. With a dissimilarity (1 - similarity) cutoff of 0.1, all people get a unique ID. If we use a cutoff of 0.4, Person 1 and 2 get the same ID, and Person 3 is assigned a different. ID. If we use a dissimilarity cutoff of 0.6, then all 3 people get the same unique ID. Some applications require one-to-one matching [**?**] [**?**], in which case we could not link Person 1, 2 and 3. In that situation we'd either need to link two or none of the three people. Please note that in this thesis we do not resolve intransitive matches or enforce a one-to-one match. Using this data to do so would be an interesting area for future work.

A.2 Using Household Information to Post-Process Individual Pairs

Another approach is to use household similarity information as a final step to assign IDs [23]. If we use a cutoff of 0.5 for our match probability cutoff, we'd need to determine whether Person 2 matches Person 1 or Person 3 since both pairs had high enough (> 0.5) individual match probabilities. From (Fig. A.2), we see that Person 2 and Person 3 have a higher household similarity than Person 2 and Person 1, so we decide that Person 2 and Person 3 get the same unique ID, and person 1 receives a separate ID.

1901	**1911**	**Similarity**	**Household similarity**
Person 1	Person 2	0.9	0.2
Person 2	Person 3	0.6	0.95
Person 1	Person 3	0.4	0.1

Figure A.2: Intransitive matching problem with a similarity cutoff of 0.5; shown with household similarity information

As in the focus of [23] we can use household information to post-process individual record pairs. For example, if we find two plausible 1911 matches for a 1901 record, we may use household information to make a decision between the two potential matches. We would do so in a process where we first build the baseline / individual model in Equation 6.4 and then use its output to calculate household similarity. Using the household similarity we can determine the matching links for ambiguous record pairs.

We build a model for individual pairs using only information about the two individuals (Line 3 of Algorithm 2). In theory we could build a model using household information (e.g., model 6.6) but if we want household information to *only* appear in the post-processing step (to avoid double utilizing this information), we should not use it in the first step. Once we have the predictions at the individual pairwise level, we determine if a pair is a match or a non-match using the cutoff c. Pairs below this threshold will be predicted to be non-matches. Pairs above are initially predicted to be matches. In this model we only want one 1911

Algorithm 2: Post-processing records using household similarity

Input: All records from 1901 and 1911 ($\boldsymbol{X}^{1901}$ and $\boldsymbol{X}^{1911}$)
Household IDs that map i to $h^{1901}(i)$ and j to $h^{1911}(j)$
Pairwise match cutoff value c

Output: The (potential) matching record for all records in $\boldsymbol{X}^{1901}$

for $i = 1, ..., n_{1901}$ **do**
  for $j = 1, ..., n_{1911}$ **do**
    Build a model for $\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911})$ with Equation 6.4 ;
    Calculate $\Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911})$;
    if $Pr(\boldsymbol{X}_i^{1901} \sim \boldsymbol{X}_j^{1911}) < c$ **then**
      Classify $\boldsymbol{X}_i^{1901}$ and $\boldsymbol{X}_j^{1911}$ as a non-match ;
    else
      Calculate $sim_{\text{group-linkage}}(\boldsymbol{X}^{1901}(h^{1901}(i)), \boldsymbol{X}^{1911}(h^{1911}(j)))$;
    end
  end
  if $j \in h_b : sim_{group\text{-}linkage}(\boldsymbol{X}^{1901}(h^{1901}(i)), \boldsymbol{X}^{1911}(h^{1911}(j)))$ *is maximum* **then**
    Classify $\boldsymbol{X}_i^{1901}$ and $\boldsymbol{X}_j^{1911}$ as a match ;
  else
    Classify $\boldsymbol{X}_i^{1901}$ and $\boldsymbol{X}_j^{1911}$ as a non-match ;
  end
end

record for each 1901 record so, if multiple 1911 records match at above the threshold c we select the record with the highest household match to the 1901 record.

Appendix B

Supplemental Models and Model Information

B.1 Model Details

B.1.1 Random Forest Variable Importance

Table B.1: RF No Blocking; Individual

	MeanDecreaseGini
locations.jar	2370.48
Age.Yea	1453.88
Forename.jar	1072.00
Surname.jar	935.12
Birthplace.jar	882.08
Sex.Exa	92.39

Table B.2: RF No Blocking; Individual + Household

	MeanDecreaseGini
Age.Yea	1545.71
locations.jar	1394.62
Forename.jar	1201.97
Forename	793.54
Age	709.65
Birthplace.jar	481.96
Surname.jar	446.27
Birthplace	443.89
Surname	422.84
Sex	401.07
Locations	247.21
Sex.Exa	145.10

Table B.3: RF No Blocking; Individual + Household (Unadjusted)

	MeanDecreaseGini
Age.Yea	1506.82
locations.jar	1482.97
Forename.jar	1239.79
Forename_	900.02
Age_	614.21
Birthplace.jar	535.95
Surname.jar	499.61
Surname_	415.31
Birthplace_	408.76
Locations_	219.66
Sex.Exa	142.02
Sex_	69.78

Table B.4: RF Block 10; Individual

	MeanDecreaseGini
locations.jar	2356.06
Age.Yea	1508.74
Forename.jar	1020.50
Surname.jar	966.66
Birthplace.jar	848.20
Sex.Exa	98.80

Table B.5: RF Block 25; Individual

	MeanDecreaseGini
locations.jar	2365.59
Age.Yea	1531.72
Surname.jar	1011.22
Forename.jar	986.44
Birthplace.jar	799.25
Sex.Exa	98.66

Table B.6: RF Block All; Individual

	MeanDecreaseGini
locations.jar	2205.87
Age.Yea	1558.62
Forename.jar	1099.80
Surname.jar	868.49
Birthplace.jar	597.61
Sex.Exa	105.46

Table B.7: RF Block 10; Individual + Household

	MeanDecreaseGini
Age.Yea	1579.76
locations.jar	1426.82
Forename.jar	1124.55
Forename	792.58
Age	626.43
Surname.jar	541.61
Surname	478.08
Birthplace.jar	471.15
Sex	417.23
Birthplace	364.29
Locations	252.34
Sex.Exa	145.46

Table B.8: RF Block 25; Individual + Household

	MeanDecreaseGini
Age.Yea	1558.16
locations.jar	1427.63
Forename.jar	1124.56
Forename	780.69
Age	595.26
Surname.jar	580.93
Surname	505.58
Birthplace.jar	478.82
Sex	405.36
Birthplace	334.51
Locations	280.29
Sex.Exa	136.40

Table B.9: RF Block All; Individual + Household

	MeanDecreaseGini
Age.Yea	1657.16
locations.jar	1334.23
Forename.jar	1281.52
Forename	758.80
Age	564.14
Surname.jar	476.69
Surname	435.40
Birthplace.jar	428.58
Sex	399.74
Birthplace	306.89
Locations	263.49
Sex.Exa	143.64

Table B.10: RF w/o Bad Labels; Individual

	MeanDecreaseGini
locations.jar	2235.97
Age.Yea	1707.60
Forename.jar	1160.14
Birthplace.jar	807.23
Surname.jar	780.58
Sex.Exa	113.83

Table B.11: RF w/o Bad Labels;Individual + Household

	MeanDecreaseGini
Age.Yea	1696.42
locations.jar	1358.93
Forename.jar	1215.24
Forename	705.28
Age	579.66
Surname.jar	509.32
Birthplace.jar	460.75
Surname	362.51
Sex	346.08
Birthplace	344.12
Locations	247.04
Sex.Exa	145.81

Table B.12: RF Page 1; Individual

	MeanDecreaseGini
locations.jar	1260.29
Birthplace.jar	306.18
Age.Yea	242.86
Surname.jar	154.57
Forename.jar	133.31
Sex.Exa	0.22

Table B.13: RF Page 1; Individual + Household

	MeanDecreaseGini
locations.jar	853.56
Forename	356.01
Locations	259.74
Age	241.60
Birthplace.jar	212.41
Age.Yea	200.29
Sex	181.60
Surname	173.48
Birthplace	155.96
Surname.jar	92.89
Forename.jar	92.66
Sex.Exa	0.33

Table B.14: RF Page 2; Individual

	MeanDecreaseGini
Age.Yea	1765.98
Forename.jar	1580.10
Surname.jar	1410.23
locations.jar	748.58
Birthplace.jar	478.36
Sex.Exa	159.83

Table B.15: RF Page 2; Individual + Household

	MeanDecreaseGini
Forename.jar	2182.35
Age.Yea	1630.44
Age	486.25
Surname.jar	443.27
Surname	313.86
Birthplace	270.56
locations.jar	263.44
Birthplace.jar	205.39
Sex.Exa	195.73
Sex	162.02
Forename	153.17
Locations	41.47

B.1.2 Linear Model Coefficients and Output

	Logreg No Blocking; Individual	Logreg No Blocking; Individual + Household
(Intercept)	-21.63***	-20.79***
	(0.23)	(0.24)
Forename.jar	5.19***	5.45***
	(0.12)	(0.12)
Age.Yea	-0.11***	-0.10***
	(0.00)	(0.00)
Sex.Exa	0.64***	0.91***
	(0.09)	(0.09)
Birthplace.jar	2.11***	1.76***
	(0.08)	(0.09)
locations.jar	6.63***	7.13***
	(0.10)	(0.15)
Surname.jar	8.29***	5.62***
	(0.15)	(0.18)
Forename		1.22***
		(0.10)
Age		1.58***
		(0.07)
Sex		-0.01
		(0.08)
Birthplace		0.12
		(0.07)
Surname		0.47***
		(0.07)
Locations		-1.37***
		(0.07)
AIC	26737.97	25201.79
BIC	26820.91	25355.82
Log Lik	-13361.99	-12587.89
Deviance	26723.97	25175.79
Num. Obs.	1033301	1033301

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table B.16

	Hier No Blocking;Individual	Hier No Blocking;Individual + Household
(Intercept)	-19.85***	-18.97***
	(0.26)	(0.28)
Forename.jar	6.59***	6.46***
	(0.14)	(0.13)
Age.Yea	-0.10***	-0.10***
	(0.00)	(0.00)
Sex.Exa	1.11***	1.13***
	(0.10)	(0.10)
Birthplace.jar	1.74***	1.43***
	(0.12)	(0.13)
locations.jar	9.10***	7.50***
	(0.20)	(0.26)
Surname.jar	2.57***	2.26***
	(0.18)	(0.17)
Forename		2.66***
		(0.18)
Age		0.48***
		(0.11)
Sex		0.40**
		(0.13)
Birthplace		0.16
		(0.12)
Surname		0.08
		(0.10)
Locations		-0.35*
		(0.16)
AIC	23059.60	22371.55
BIC	23154.39	22537.43
Log Lik	-11521.80	-11171.78
Num. Obs.	1033301	1033301
Groups: House	37972	37972
Var: Group(Intercept)	5.14	4.08

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table B.17

B.2 Modeling Households Directly

Because we collect whether households match (0/1) we could also directly model whether the households are the same.

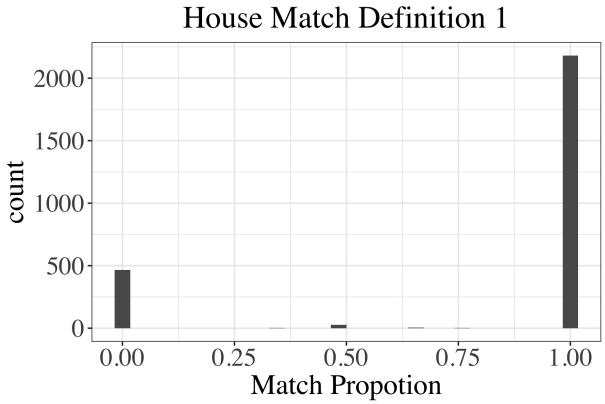


Figure B.1

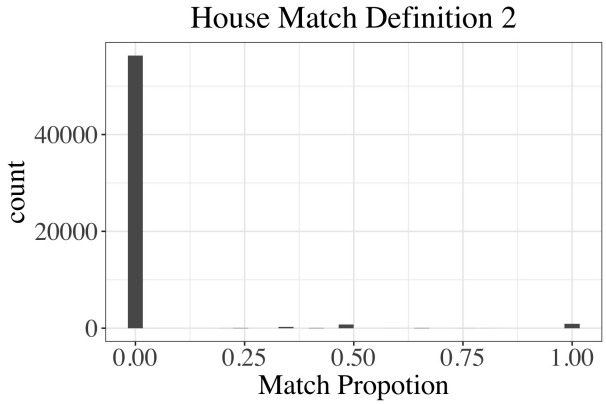


Figure B.2

$$\Pr(\boldsymbol{X}^{1901}(h^{1901}(i)) \sim \boldsymbol{X}^{1911}(h^{1911}(j))) = f\big(sim(\boldsymbol{X}^{1901}(h_a),\, \boldsymbol{X}^{1911}(h^{1911}(j)))\big) \tag{B.1}$$

Table B.18: Household Definition 1

	MeanDecreaseGini
Age	71.10
Sex	64.45
Forename	61.99
Surname	59.26
Birthplace	32.52
locations	13.96

Table B.19: Household Definition 2

	MeanDecreaseGini
Forename	308.75
locations	282.86
Age	232.71
Surname	143.64
Birthplace	140.58
Sex	132.86

B.3 Other Graphs

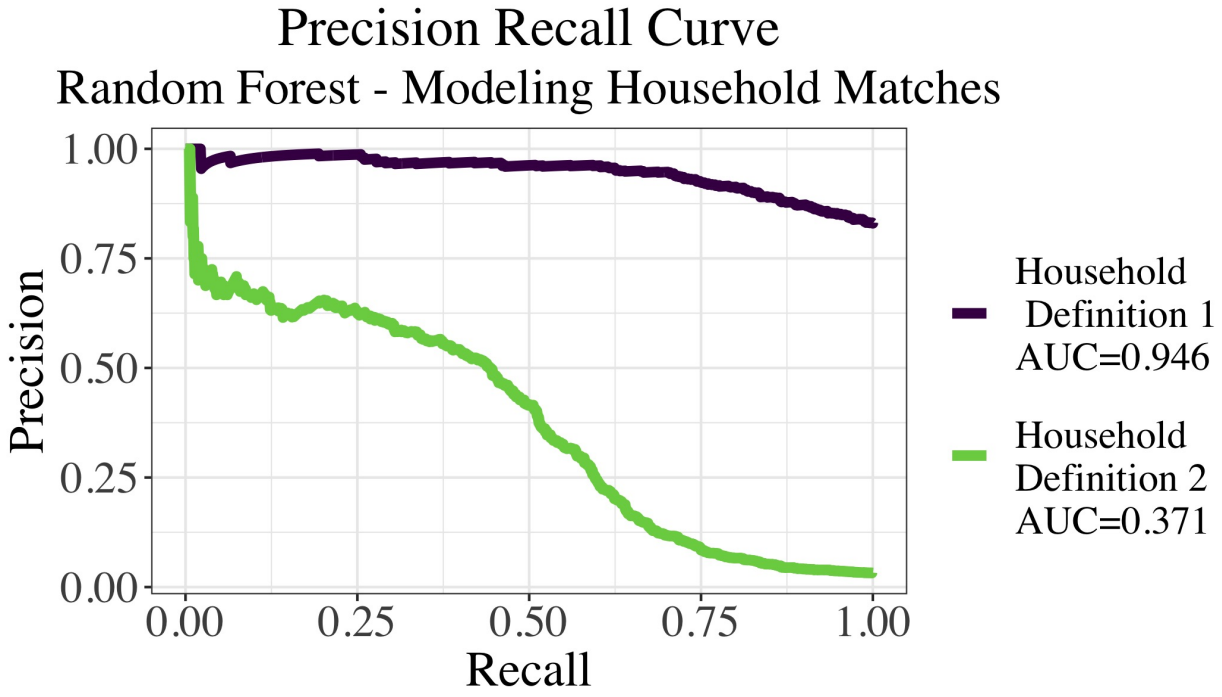

Precision Recall Curve
Random Forest - Modeling Household Matches
Precision
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Recall
Household Definition 1 AUC=0.946
Household Definition 2 AUC=0.371

Figure B.3

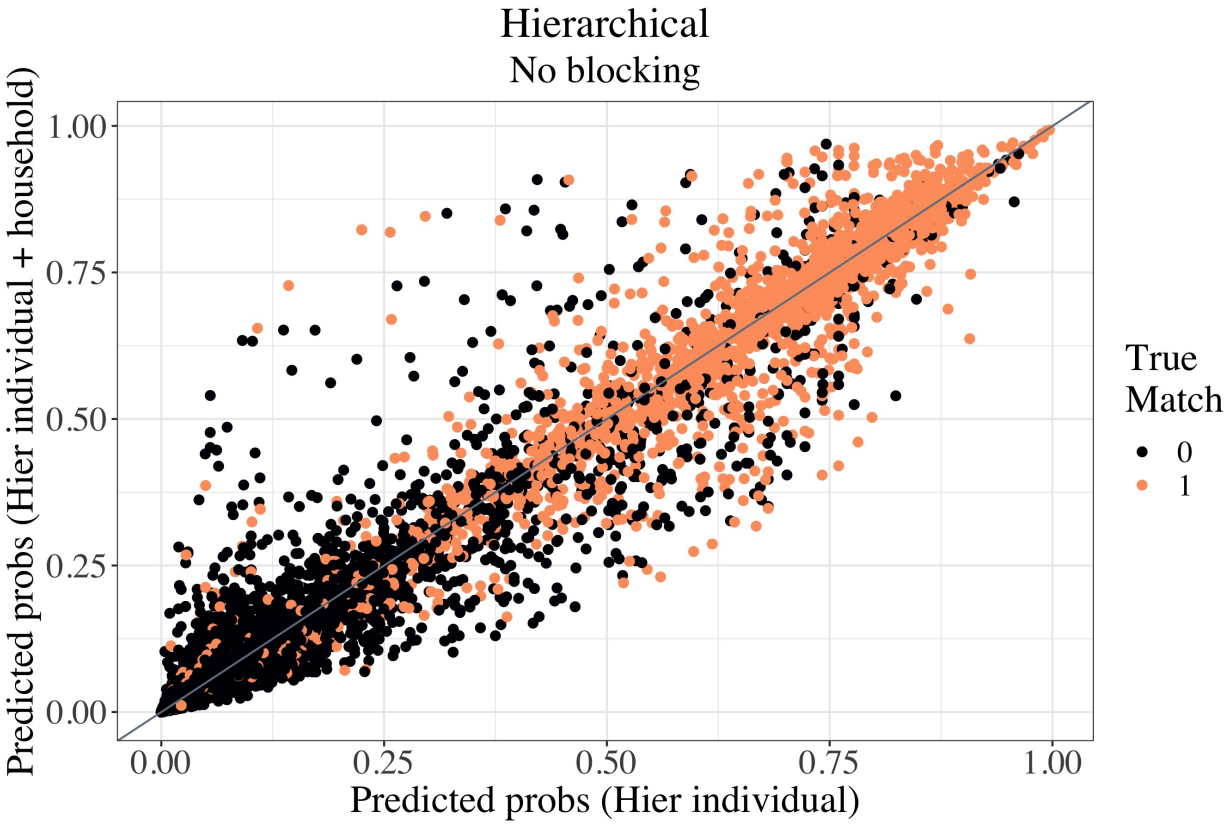

Hierarchical
No blocking
Predicted probs (Hier individual + household)
Predicted probs (Hier individual)
0.00
0.25
0.50
0.75
1.00
True
Match
0
1

Figure B.4

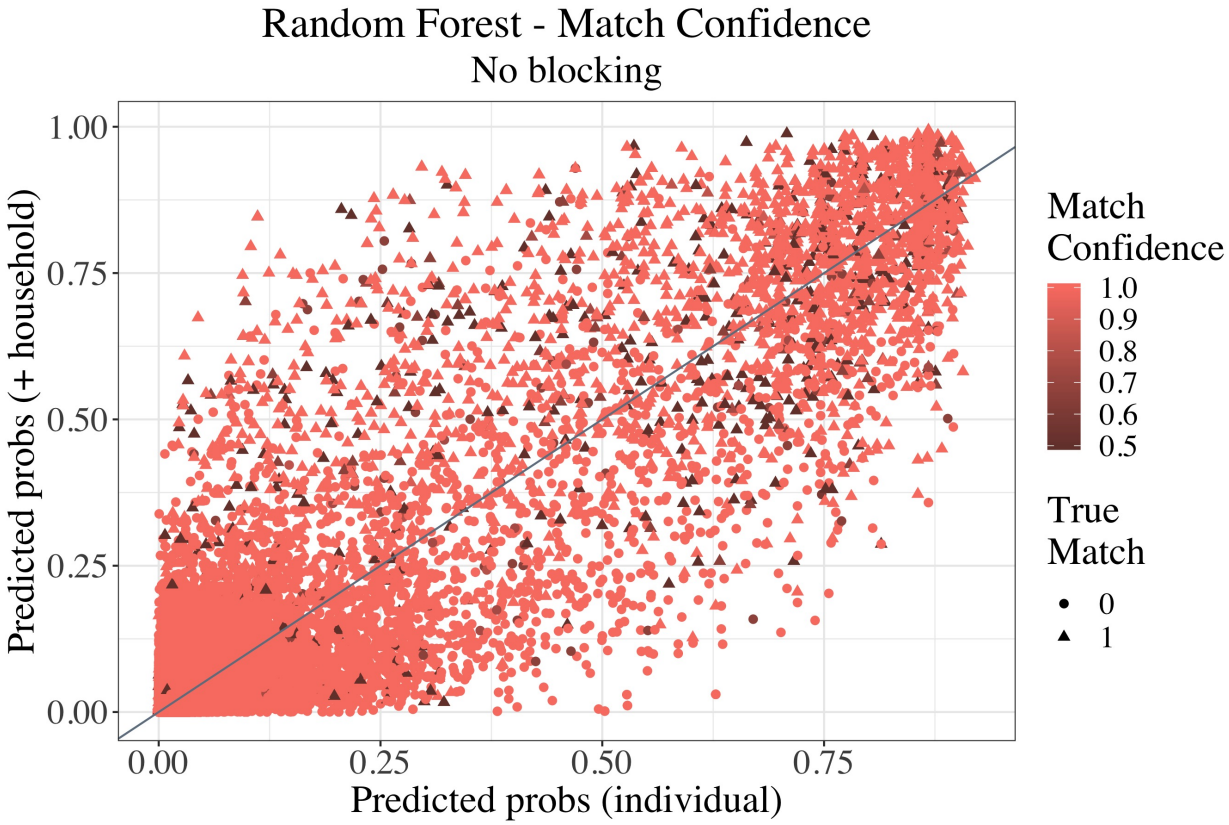

Random Forest - Match Confidence
No blocking
Predicted probs (+ household)
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
Predicted probs (individual)
Match Confidence
1.0
0.9
0.8
0.7
0.6
0.5
True Match
0
1

Figure B.5

Appendix C

Supplemental Interface Analyses

While the main goal was to collect the additional matching individuals, we are also interested in studying how people label records. Labelers are shown a 1901 reference record, and a set of 1911 candidates. These candidates are of varying length, and chosen based on a combination of geography and field similarity. On this first page (where a labeler decides if the reference matches to any of the candidates) we record whether the labeler viewed the household members, which candidates were viewed and in what order. Additionally we store whether they sorted the fields (if so, which ones? how often?) and the time stamps of these actions.

We found that our labelers often only looked at a few candidates, but sometimes they clicked on as many as 22 candidates when searching to determine if there is a correct match. The number of clicks is inherently a function of the number of candidates shown, as we see in Fig. C.1. We understand that there is also a labeler effect/component in that some people tend to click around more than others and are naturally more/less decisive.

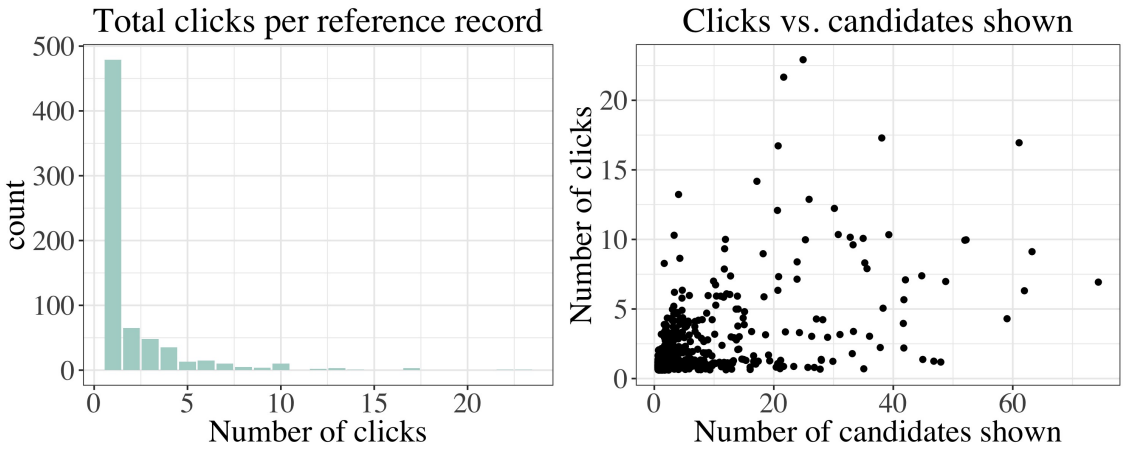


Figure C.1: Labelers often only looked at a few candidates, but sometimes clicked on as many as 22 candidates (left). The number of clicks is a function of the number of candidates shown as well as other variations (right).

C.1 Assessing the uncertainty of human-based record linkage

Another benefit of the interface is the ability to easily collect label uncertainty. By exposing multiple labelers to the same reference record, we can assess the uncertainty of the link which can help downstream record linkage models avoid overfitting to uncertain matches. Currently, each record has between one to fifteen recorded labels and references have between one to four unique labels showing that there indeed exists disagreement between labelers.

Early assessment of labels suggest some uncertainty between labelers, which begs the question, how many labels are sufficient (in terms of labeler variability) to stop labeling? How do we balance the cost and benefit associated with generating a label? In Fig. C.2 we track how the max candidate proportion changes as we generate more and more label instances for a particular reference record. We call the most common label for a given reference record the "max candidate". Excluding those references that had fewer than six label instances or no disagreement amongst labelers (76% of reference records had no disagreement amongst labelers), in Fig. C.2 we find that the sequences which bounce around with low label agreement tend to be green and blue (labeler disagreement in the first two or three instances respectively), indicating that extreme uncertainties are associated with early inconsistencies. This is an important preliminary finding that will be further analyzed with future crowdsourcing.

C.2 Understanding labeler decisions through click patterns

Looking beyond label variability, understanding how humans label records will help inform how best to build record linkage models and labeling interfaces (including our own). Preliminary analysis of the labelers' exploration of candidates (specifically the sequences of labeler clicks as they explore candidates), suggest some potential in this area. For example, the reference record James Byrne had 33 possible candidates, but looking at the click sequence for multiple labelers we see that they primarily focused on the first few candidates although the candidates were shown in random order. The three candidate ID click sequences for James were: 7_3_7_7, 7_2_3_4_12_2_7, and 2_7.

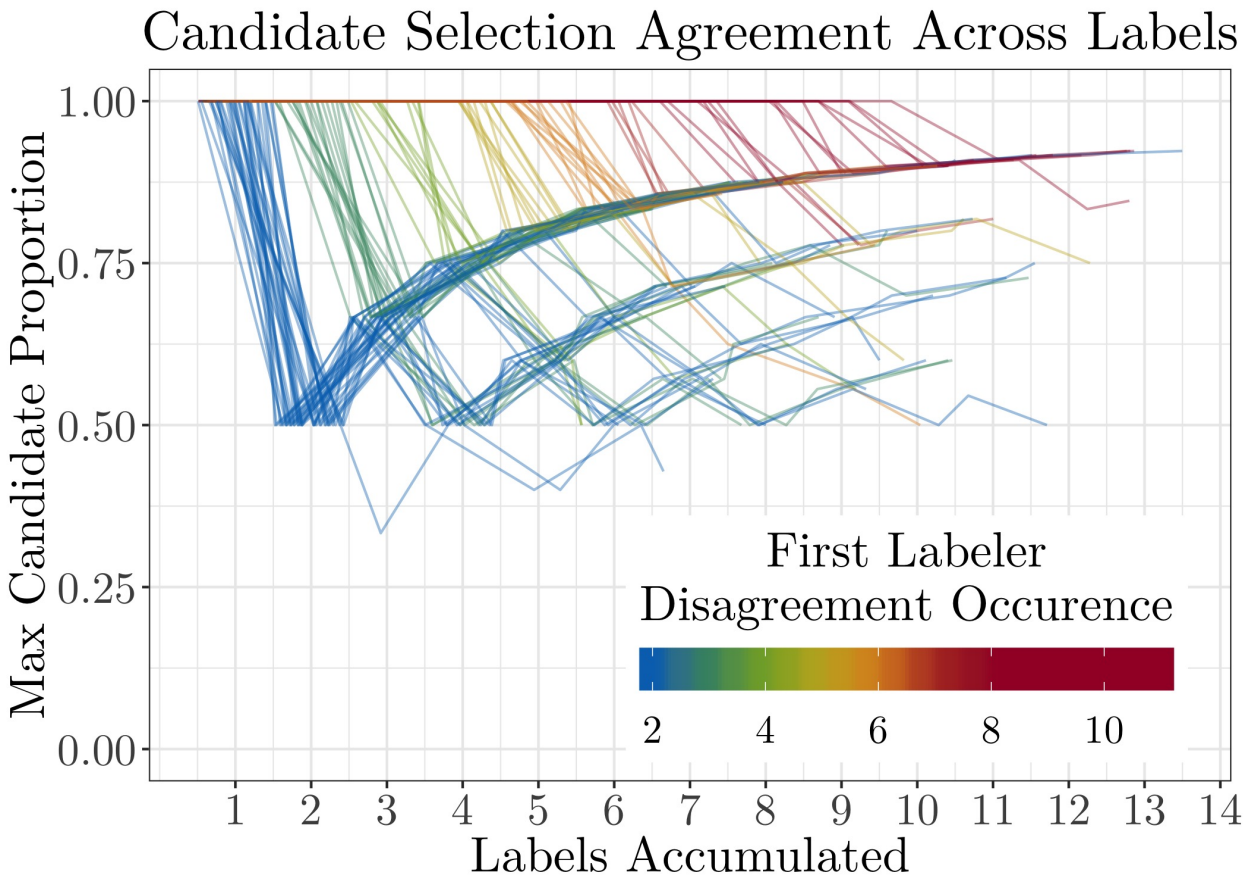


Figure C.2: We examine how the max candidate proportion changes as we generate more label instances for a particular reference record. We call the most common label for a given reference record the "max candidate". One line in this graph represents a reference record that has at least six label instances and whose final match proportion is less than 1 (some uncertainty exists). The lines are colored by the first time that a labeler disagrees with previous labelers (e.g., the second color from the left – blue-green – shows those sequences where the first two labelers agreed on the candidate, but the third labeler did not). We find that the sequences that finish with low label agreement tend to be green and blue, indicating that future label uncertainty is associated with early label inconsistencies.

Moreover, Fig. C.3 shows that the way that labelers explore potential candidates may have semi-cyclic patterns in terms of increasing and decreasing the need for candidates that have high individual and / or household similarity scores. We believe this information could enhance record linkage models and our

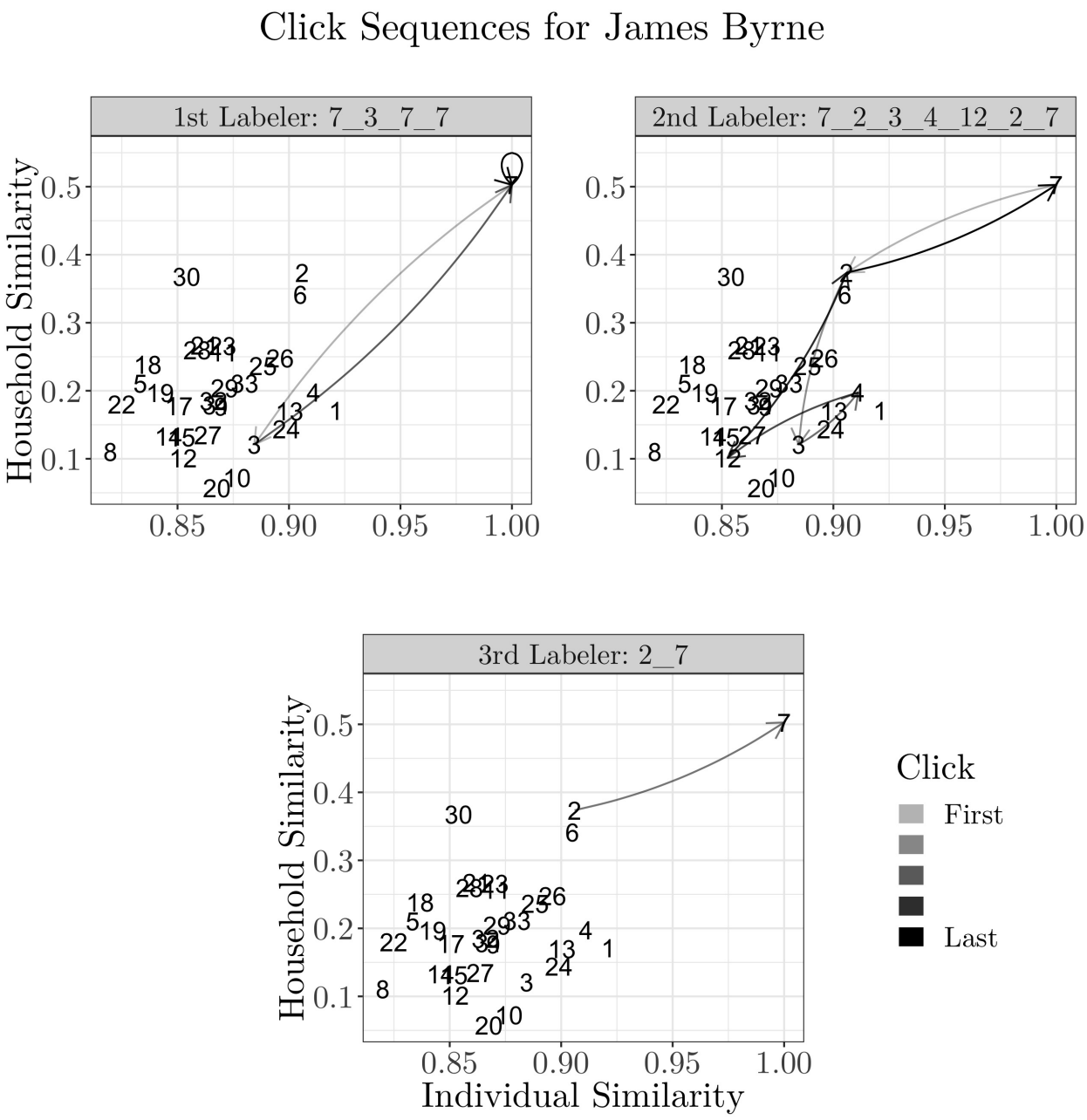


Figure C.3: We examine the click sequences of three labelers as they explore potential candidates of 1901 James Byrne. This visual examines the labelers' exploration sequences relative to the similarity of the reference / candidate individual records as well as their respective households.

candidate selection process by helping us learn which candidates appear to be most important to a human labeler.

Appendix D

Unsupervised Linkage

Note: This section is largely part of [22]. This is using data that had been collected at the very early stages of our interface and therefore do not include labels for the households or individuals within households. Geographically, the data in this section is from Ticknock, County Carlow.

D.1 Fellegi & Sunter

In the absence of representative labels or a reluctance to rely on estimated labels, we turn to exploring unsupervised approaches, starting with the fundamental work by Fellegi & Sunter [20]. This approach was later expanded by Winkler [**?**].

For records a, b, and each binary comparison vector, $\tau(a, b)$ of length K (the number of variables/fields), we calculate the following likelihood ratio:

$$R(\tau(a,b)) = \frac{\mathcal{P}(\tau|\text{ true match})}{\mathcal{P}(\tau|\text{ true non-match})}. \tag{D.1}$$

We make decisions about the match status of pairs of records based on cutoffs where a high $R(\tau(a, b))$ indicates that a and b likely match and a low $R(\tau(a, b))$ indicates that a and b likely do *not* match.

In practice there is often a clerical review step for record pairs with neither high nor low ratios to determine their match status, but here we set a hard binary cutoff for determining matches and non-matches. The Fellegi & Sunter approach has been shown to minimize both false positive/negative linkage errors and the number of possible links. With respect to estimation, the goal is to maximize the following likelihood $f(\tau)$ for each comparison τ:

$$
\begin{aligned}
f(\tau) &= \mathcal{P}(\text{match})\mathcal{P}(\tau|\text{match}) + \mathcal{P}(\text{non-match})\mathcal{P}(\tau|\text{non-match}) \\
&= p\prod_{k=1}^{K} m_k^{\tau_k}(1-m_k)^{(1-\tau_k)} + (1-p)\prod_{k=1}^{K} u_k^{\tau_k}(1-u_k)^{(1-\tau_k)}
\end{aligned} \tag{D.2}
$$

with parameters:

$$
\begin{aligned}
m_k &= \mathcal{P}(a \text{ and } b \text{ agree on variable } k \mid \text{true match}) \\
u_k &= \mathcal{P}(a \text{ and } b \text{ agree on variable } k \mid \text{true non-match}) \\
p &= \mathcal{P}(a \text{ and } b \text{ are a match })
\end{aligned}
$$

E-M algorithm

The E-M algorithm is often used for parameter estimation of the Fellegi & Sunter approach[**?**]. Note that this approach commonly uses binary comparison vectors where each element indicates whether or not two fields agree/match (vs. a more continuous similarity score). Using our estimated Ticknock labels as ground truth, we determined reasonable cutoffs for our continuous similarity metrics, finding that a Jaro-Winkler cutoff of 0.75 fairly well separates the matching and non-matching pairs. With respect to age, recall that we expect variation beyond the expected ten-year age gap given the national change in benefits during that time period and typical errors in transcription or record-keeping. We are able to capture most of the true matching pairs when we allow the age difference to vary between 6 and 14 years. Based on this preliminary sensitivity analysis, we use the following criteria to dichotomize our variables. We use a Jaro-Winkler cutoff of 0.75 for birthplace, location, first name, and last name. We allow age to differ 4 years from the expected age difference of 10 years, and we require gender to match exactly.

We use the following variables in the Fellegi & Sunter model:

Variables: Forename-JW > 0.75, Surname-JW > 0.75, Birthplace-JW > 0.75, Location-JW > 0.75, $6 <$ Age-Diff < 14, Sex-Exact.

Parameter estimation

We first look at the E-M iteration convergence results for estimating p, the probability that a given comparison vector corresponds to a true match. Note that for this analysis of 331 records from 1901 and 313 records from 1911, the true proportion of matches is 204/105,118 = 0.00194. We find, in Table D.1, that our algorithm quickly overestimates the number of true matches. This is perhaps somewhat expected given the very low overall percent of matches.

Table D.1: Estimating p

Iteration 1	Iteration 15	Iteration 30	Truth
0.00031	0.01195	0.06292	0.00194

We similarly compare the estimates for m_k and u_k across E-M iterations. We expect most of our m_k probabilities to be high; if two people are truly the same person, we typically expect them to agree on most variables. As such, we initialize m_k to be 0.95 for all fields. On the other hand, we expect our u_k probabilities to vary depending on the field of interest. For example, even among non-matching people, we expect gender to agree about 50% of the time. We expect the u-probability for age to be higher than say, last name, but not as high as gender. We initialize each u_k to be the total number of agreeing comparisons for field k over the total number of comparisons. This initialization is reasonable given the large class imbalance in our comparison space.

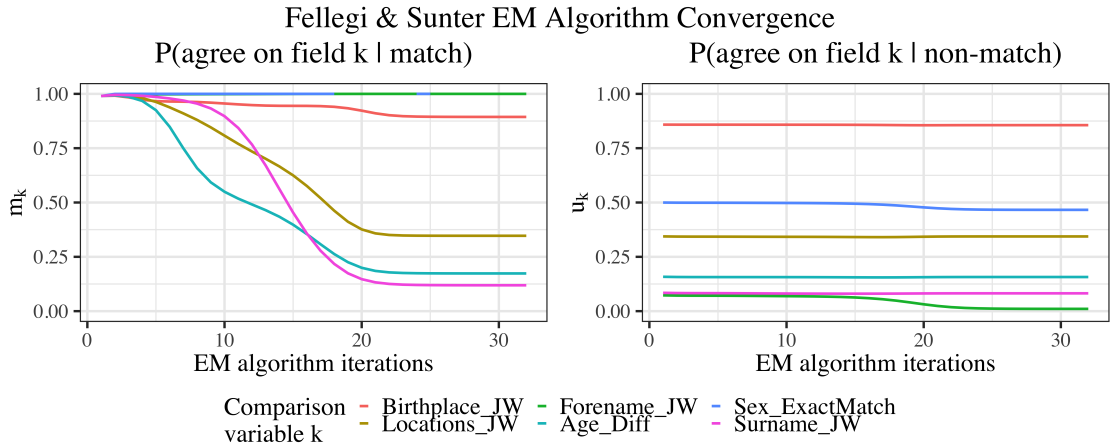


Figure D.1: We explore the convergence of m_k and u_k within our model and examine the difference between their final values for each field. A large difference between m_k and u_k for a given k indicates that field k is important in separating matches from non-matches. In the individual model we find that first name and sex are driving our likelihood ratios, but other fields like last name, location, and age are not playing a strong role.

The convergence of these parameters is shown in Fig. D.1. A large difference between m_k and u_k for a given field k indicates that the variable is influential in determining the match status of the record pair. For Ticknock, we find that first name and sex are driving our comparison vector likelihood ratios (D.1), but, contrary to intuition, other fields like last name, location, and age are not playing a strong role.

D.2 Building Unsupervised Models

Given this household similarity measure, we explore how to add this information to our Fellegi & Sunter model, keeping in mind that our end goal is still to link individual records. We will assess the following approaches:

1. Baseline model (no household information)

   Fellegi-Sunter(∼Forename-JW + Surname-JW + Sex-Exact+
   Location-JW + Age-Abs-Diff + Birthplace-JW)

2. Add household field similarities to the baseline model

   Fellegi-Sunter(∼[Model 1 variables] +
   Forename-House-Jac + Surname-House-Jac + Sex-House-Jac+
   Location-House-Jac + Age-House-Jac + Birthplace-House-Jac)

3. Add estimated household similarity to the baseline model

   Fellegi-Sunter(∼[Model 1 variables] +
   Household-Similarity)

4. Estimate a Fellegi & Sunter model on the households to determine which records should be compared in the baseline model

   Stage 1 (blocking):

   Fellegi-Sunter(∼[Household-derived variables])

   Stage 2:

   Fellegi-Sunter(∼[Model 1 variables])

In option 2, we calculated the adjusted Jaccard similarity between all households for the fields: forename, age (binned), surname, birthplace, sex, and location. We then added those household similarity variables to

the individual field comparisons (Jaro-Winkler for birthplace, forename, surname, and location; exact match for gender; expected difference +/- 4 years for age) in our individual-level model. Results are discussed below and shown in the upper right graph of Fig. D.2).

In option 3, we ran two Fellegi & Sunter models sequentially. First we modeled the individual matches (e.g. person 1 matches/non-matches person 2) and then we used those matches to determine the household similarity (group linkage measure, shown in equation 5.16). We then model the individual matches again, but include the household similarity as a feature in the model.

In option 4, we ran two Fellegi & Sunter models sequentially. First we modeled household matches (e.g. household 1 matches/non-matches household 2) and then we modeled the individuals (e.g. person 1 matches/non-matches person 2). We used the likelihood ratios from the household model to determine whether two households were similar enough to compare the people within them, effectively blocking our records based on household similarity.

We also explored including the Fellegi & Sunter likelihood ratio for household comparisons as a variable in the individual-level model, but the performance was comparable to the poor performance of the individual-level model with the larger set of household similarity variables.

Given the introduction of an additional likelihood ratio cutoff selection, we explore, in Table D.3, the sensitivity of error rates as a function of the household match threshold. The top line shows our individual-level model, with no household level blocking. As we reduce our comparison space with the increase of the household match threshold, the false negative error rates increase (as expected). Ultimately, we set a household blocking ratio cutoff of 2, reducing the comparison space from 105118 record pairs to 24298.

Model	False Negative	False Discovery
1 (baseline)	0.24	0.23
2 (+ household vars)	0.78	0.81
3 (+ household sim)	0.27	0.17
4 (household FS to block + baseline FS)	0.32	0.17

Table D.2: Error rates (varying likelihood ratio cutoffs)

In Fig. D.2 we compare all four approaches as a function of the likelihood ratio cutoff for the individual level matches and non-matches. Including household variables in the baseline model (option 2) increased our error rates; however, including the group linkage household similarity (option 3) produced similar error rates to the baseline model (higher false negative, lower false discovery). When using option 3 we are determining more matches to be non-matches but we are less often incorrectly predicting matches of non-matching pairs. Estimating household similarity as a blocking mechanism for the individual-level model (option 4) resulted in a slightly worse performance than the baseline model. In addition to the reduction of the comparison

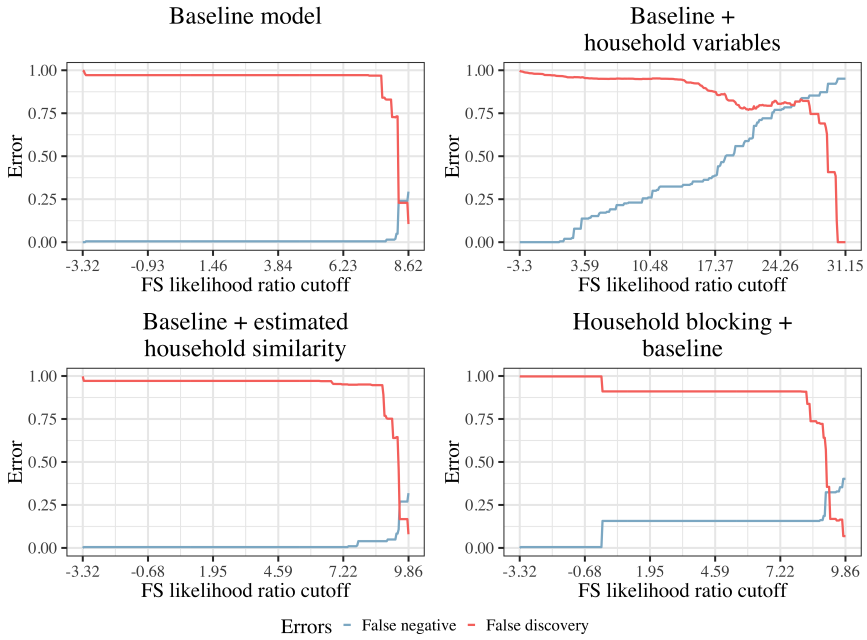


Figure D.2: We compare the error rates of different models that incorporate household information to the baseline unsupervised model. The model that includes household-level variables into the individual model (top right) performs poorly. The model that includes the group linkage estimated household similarity (bottom left) performs comparable to the baseline model. Additionally, the model that uses household information to block (right) performs comparable, but slightly worse than the original individual model.

space, we find the m and u probabilities for the household model are more reasonable than the individual model (Fig. D.3).

D.2.1 Modeling Household Similarity Directly

In Option 4, we first modeled household matches. We were able to do this because we were using an unsupervised approach (at this point, we do not know whether or not two households actually match). The m and u probabilities for the household model are shown in Fig. D.3. The adjusted Jaccard similarity for gender and birthplace appear to be the most influential. While not large, we still see a difference between m_k, u_k for the household-level comparisons of forename, surname and age. Note that the influential fields here are different than those in the individual-level model (Fig. D.1). We find slightly more reasonable m and u probabilities, as shown in Fig. D.3. For example, age is now more influential in determining matches.

Given the introduction of an additional likelihood ratio cutoff selection, we explore, in Table D.3, the sensitivity of error rates as a function of the household match threshold. The top line shows our individual-level model, with no household level blocking. As we reduce our comparison space with the increase of the household match threshold, the false negative error rates increase (as expected). Ultimately, we set a household blocking ratio cutoff of 2, reducing the comparison space from 105118 record pairs to 24298.

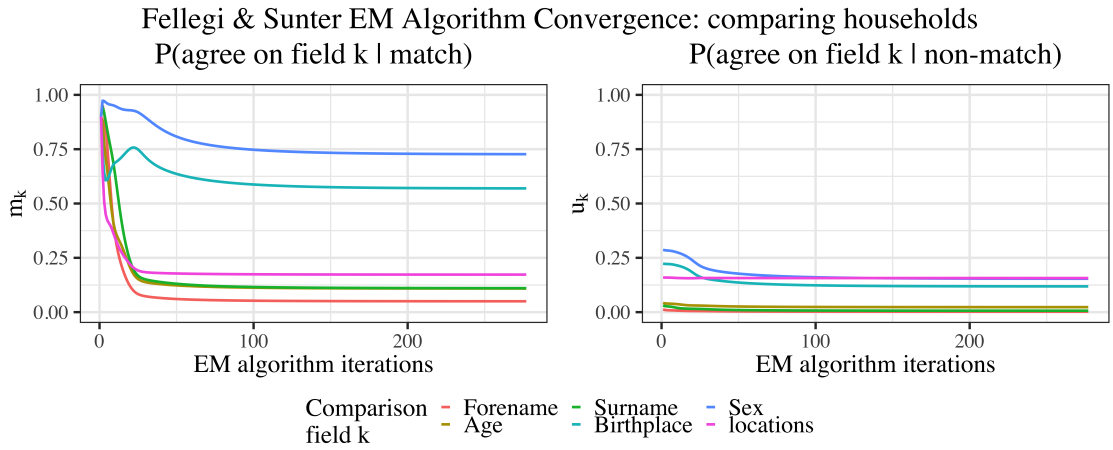


Figure D.3: Convergence of m_k and u_k within the household matching model. In this model we find that the adjusted Jaccard similarity for birthplace and sex across the households are driving our likelihood ratios.

Table D.3: Households to block: comparison space reduction

FS ratio cutoff	n	False Discovery	False Negative
-Inf	105118	0.35	0.24
-2.10	65376	0.32	0.27
0.50	49464	0.32	0.28
1.00	28611	0.28	0.28
2.00	24298	0.25	0.32
4.00	22609	0.26	0.34
6.50	2419	0.22	0.48

Vita

Vita goes here...