Linking Historical Census Data Across Time

Zhichun Fu

A thesis submitted for the degree of
Doctor of Philosophy of Computer Science
The Australian National University

May 2014

Except where otherwise indicated, this thesis is my own original work.

Zhichun Fu
22 May 2014

Dedications

to

the ones I love

Acknowledgments

I would like to take this opportunity to thank my supervisor, Associate Professor Peter Christen, for his guidance, support, encouragement, and patience throughout my PhD studies. His insight, breadth of knowledge and enthusiasm has been invaluable for me. Without his instruction and help, supervision and friendship, I would not be able to complete this work.

I would also like to thanks Dr. Mac Boot from the Australian Demographic and Social Research Institute, College of Arts and Social Science in the Australian National University, for providing me with the data sets used in my research. His knowledge in social science and historical census helps me a lot, especially in the data analysis step.

Thirdly, I would like to thanks Dr. Jun Zhou from the School of Information and Communication Technology at Griffith University for his help in providing ideas to solve many theoretical and technical problems in my studies. His encouragement is highly important in inspiring me in both life and research.

Last but not least, I would like to give special thanks to Dr.Tiberio Caetano from NICTA for his support of my research. I also want to show my gratitude to my family and friends for standing behind me during the tough times and always giving me kind words of encouragements.

Abstract

Historical census data provide a snapshot of the era when our ancestors lived. Such data contain valuable information for the reconstruction of households and the tracking of family changes across time, which can be used for a variety of social science research projects. As valuable as they are, these data provide only snapshots of the main characteristics of the stock of a population. To capture household changes requires that we link person by person and household by household from one census to the next over a series of censuses. Once linked together, the census data are greatly enhanced in value. Development of an automatic or semi-automatic linking procedure will significantly relieve social scientists from the tedious task of manually linking individuals, families, and households, and can lead to an improvement of their productivity.

In this thesis, a systematic solution is proposed for linking historical census data that integrates data cleaning and standardisation, as well as record and household linkage over consecutive censuses. This solution consists of several data pre-processing, machine learning, and data mining methods that address different aspects of the historical census data linkage problem. A common property of these methods is that they all adopt a strategy to consider a household as an entity, and use the whole of household information to improve the effectiveness of data cleaning and the accuracy of record and household linkage.

We first proposal an approach for automatic cleaning and linking using domain knowledge. The core idea is to use household information in both the cleaning and linking steps, so that records that contain errors and variations can be cleaned and standardised and the number of wrongly linked records can be reduced. Second, we introduce a group linking method into household linkage, which enables tracking of the majority of members in a household over a period of time. The proposed method is based on the outcome of the record linkage step using either a similarity based method or a machine learning approach. A group linking method is then applied, aiming to reduce ambiguity of multiple household linkages. Third, we introduce a graph-based method to link households, which takes the structural relationship between household members into consideration. Based on the results of linking individual records, our method builds a graph for each household, so that the matches of households in different census are determined by both attribute relationship and record similarities. This allows household similarities be more accurately calculated. Finally, we describe an instance classification method based on a multiple instance learning method. This allows an integrated solution to link both households and individual records at the same time. Our method treats group links as bags and

individual record links as instances. We extend multiple instance learning from bag to instance classification in order to allow the reconstruction of bags from candidate instances. The classified bag and instance samples lead to a significant reduction in multiple group links, thereby improving the overall quality of linked data.

Contents

List of Figures

List of Tables

List of Notations

T	Total number of datasets
$\mathcal{D} = \{\mathcal{D}_t\}$	Historical census datasets, $1 \leq t \leq T$
$H_{t,i} \in \mathcal{D}_t$	i^{th} household in dataset $\mathcal{D}_t$
$r_{t,j} \in \mathcal{D}_t$	j^{th} record in dataset $\mathcal{D}_t$
$R_t = \{r_{t,j}\}$	Set of all records in dataset $\mathcal{D}_t$, $1 \leq j \leq M_t$
$H_t = \{H_{t,i}\}$	Set of all households in dataset $\mathcal{D}_t$, $1 \leq i \leq N_t$
$r_{t,i,j} \in \mathcal{D}_t$	j^{th} record in the i^{th} household of dataset $\mathcal{D}_t$
M_t	Total number of records in dataset $\mathcal{D}_t$, $M_t =\mid R_t \mid$
N_t	Total number of households in dataset D_t, $N_t =\mid H_t \mid$
$M_{t,i}$	Number of records in household $h_{t,i}$, $M_{t,i} =\mid h_{t,i} \mid$
$Rs(r_{t,i,j}, r_{t',i',j'})$	A vector containing attribute-wise similarities for record pair $r_{t,i,j}$ and $r_{t',i',j'}$
$Rsim(r_{t,i,j}, r_{t',i',j'})$	Overall similarity score for record pair $r_{t,i,j}$ and $r_{t',i',j'}$
$Hsim(H_{t,i}, H_{t',i'})$	Household pair similarity score for household $H_{t,i}$ and $H_{t',i'}$
ρ	Cut-off threshold to decide whether two records match, $\rho >= 0$
$\mathcal{M}$	Matched record pairs between two households
$\mathcal{Q}$	$\mathcal{Q}$ is the number of similarities generated from different approximate string matching methods on the record attributes.
$G = (V, E, \alpha, \beta)$	Attributed graph
V	A set of vertices
E	A set of edges, $E \in V \times V$
α	Vertex attributes
β	Edge attributes
$GSim(h_{t,j}, h_{t',j'})$	Graph similarity score, between households $h_{t,j}$ and $h_{t',j'}$

$B = \{B+, B-\}$	Bags in multiple instance learning. $B+$ and $B-$ are positive bags and negative bags, respectively. A bag contains a set of linked record pairs between two households from different datasets.
$MSim(B_k, r_{t,i})$	Similarity between bag B_k and record $r_{t,i}$ in multiple instance learning method.

Chapter 1

Introduction

Historical census data captures information about our ancestors. They help social scientists to understand how people lived, as well as the economic, social, and demographic features of their society [1]. Crucial as they are, census returns are still only snapshots of moments in time. The value of these snapshots is greatly enhanced, however, if they can be linked to the same individuals, families, and households over several censuses. Linked census data can provide researchers with new insights into the dynamic character of social, economic, and demographic change; enable researchers to reconstruct the key life course events of large numbers of individuals, households, and families; and ask new questions about changes in society and its history at levels of detail far beyond the scope of traditional methods of historical research [2, 3]. They may even facilitate epidemiological studies of the genetic factors of diseases such as cancer, diabetes, or mental illnesses [4].

In the past, social scientists have linked census records manually. Due to the amount of data to be processed and the complexity of the task, this process is very expensive in terms of both time and human resources, and such exercises are usually restricted to small numbers of individuals and households over short periods of time. In order to relieve social scientists of the tedious task of manual linkage, there are strong needs to develop automatic or semi-automatic data linkage techniques. This will allow them to concentrate their time and efforts on the actual analytic research and writing-up of results.

Various automatic or semi-automatic historical census data linking methods have been explored by computer science researchers and social scientists [2, 3, 4, 5, 6, 7, 8]. Some techniques have applied string comparison techniques to match individuals, some have linked historical census data with other types of data, and others have used Bayesian inference or discriminative learning methods to distinguish matched (two records correspond to the same entity) from non-matched (two records do not correspond to the same entity) records [9]. A detailed overview of current applications is provided in Chapter 2. Although progress has been made in this area, current solutions are far from practical in terms of both accuracy and efficiency in dealing with historical census datasets. There are significant needs in developing effective methods to link historical census data across time. Our goal in our research is to provide social scientists with a set of tools that facilitates automatic historical census data cleaning, standardisation, and linkage.

1.1 Challenges

Historical census data linkage is a nontrivial task because of several reasons, including poor data quality, large amount of common values in certain record attributes, limited and non-standard information, and population dynamics.

First, the poor quality of the original census returns and of some modern transcriptions is notorious. Large amounts of errors and inaccurate information have been introduced into the data during the census collection and digitisation processes [2]. The first type of errors are due to the low levels of literacy and education of household members. The census return instruction confused many householders and so they were unable to complete the returns correctly. Then new errors were introduced when data was transcribed from the schedules onto enumerator return sheets and in the consolidation stages. Finally, errors happened when the enumerator returns were digitised into digital images, when key strokes of handwriting became incomplete,

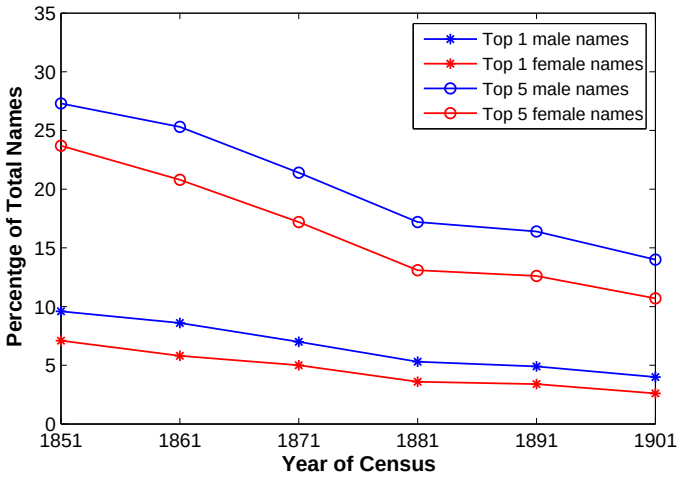


Figure 1.1: Percentage of top male and female names in six census datasets used in our research.

or even disappeared. The combination of these three types of errors, when combined together, lower the chance of correct data linkage.

Second, a large portion of records commonly contain the same or similar values in certain attributes. This is especially the case for name and address attributes. In every decade, some popular first names are given to newborns by a large percentage of a population. In Figure 1.1, we give a summary of the percentage of the top 1 and top 5 names that have appeared in the six historical census datasets from the UK that are used in our research. It can be seen that the top 5 names form a significant proportion of all first names. This implies a large number of records will have highly similar values in the first name similarity calculation from historical census data.

In early census data, many addresses do not have street numbers. This is either because houses were not numbered in a street, or because this attribute was not included in the census. Therefore, the addresses of people living in the same street become the same. It is not uncommon to find records of different people with the same name, the same age, and living in the same street, in one dataset.

Third, there is no standard format in some key attributes. In the census data used in our research, the “AGE" attribute does not only contain numbers to express a person’s current age in years, but it also contains some other values such as “under 1 m", “1/2 m", and “< 1 m". The “FIRST NAME" attribute is sometimes combined with a middle name, while the “OCCUPATION" attribute has different expressions for the same occupation.

Fourth, the structure of households and their members can change significantly between two censuses. People may be born or die, which makes the sizes of households increase or decrease. When people get married, new members may join a household, while in other cases, children move out after marriage. People may change occupation or move address. Guests or servants may be in a household on the census night. On one hand, such population dynamics reflect key information on how a society changes, which is one of the most important types of information to be extracted from census data. On the other hand, population dynamics has greatly increased the uncertainty of the data and the difficulty of linking records or households across censuses.

Fifth, in historical census data collected in the 19th and early 20th century, only limited information about individuals and household is available. For example, in the historical census data used in our research, only 12 attributes have been included. Even so, many of them are often left blank or contain no meaningful values. Modern census data, such as the 2011 Australian Census[1], contain 60 questions in the census form, which could make the task of linking records and households much easier.

Given these challenges of poor data quality, presence of limited and non-standard formats in key attributes (fields), complications of common names and identical

[1] http://www.abs.gov.au/ausstats/abs@.nsf/Lookup/2901.0Main%20Features802011

addresses, opportunities for population dynamics to transform populations significantly between censuses, and the limited information collected in the early ages, it is not surprising to find that attempts at using automatic linkage techniques to deal with large collections of census data have been disappointing [10].

1.2 Motivation

Many methods have been proposed in order to address the challenging historical census data linkage problem. The main efforts have been on developing data cleaning and standardisation approaches to improve data quality, and using various string comparison methods for more reliable estimation of attribute similarity. Nonetheless, the problems of limited attributes, population dynamics, and common names and identical addresses are not so easy to be solved. The difficulty of eliminating ambiguous links, i.e., several records from one dataset linked to one record in another dataset, still remains as the major obstacle for highly accurate data linkage.

In order to develop highly accurate historical census data linkage, three questions need to be answered.

- What is the information that can be extracted from historical census data to facilitate the linkage task?
- What are the data mining and pattern recognition approaches that can be used to increase accuracy?
- How to extend existed approaches, so that new methods can be developed to improve the linkage accuracy of historical census data linkage?

The answer to the first question comes from domain knowledge of historical census data. When census data were collected, the answer sheets were filled by householders. Therefore, a household shall be considered as an integrated entity. This is also

how social scientists link individual records manually. They take the household context into consideration. This enables tracking of the majority of members in a household over a certain period of time, which facilitates the extraction of information that is hidden in the data, such as fertility, occupations, changes in household structures, immigration and movements, and so on. Such information normally cannot be easily acquired by only linking records that correspond to individuals. Therefore, the first motivation of this research is: *we should explore household information for historical census data linkage*.

In accordance with the first motivation, and to address the second and third questions above, we propose the second motivation of this PhD research as: *we should develop data linkage methods that can treat a group of records as an entity*. Note that most data linking methods have been developed with the aim to match records from individuals, while the internal relationship between household members has been ignored. When a group of records is considered as a whole entity, novel methods can be developed to explore such internal relationships for each step of the data linkage process. These steps include data analysis, data cleaning and standardisation, and data linkage and classification. In particular, the latest development of data mining, machine learning, and pattern recognition shall be explored, extended, and exploited, to meet the requirements of historical census data linkage.

1.3 Contribution of the Thesis

The ultimate goal of linking historical census data in efficient and effective ways is to provide social scientists with tools to reconstruct various aspects of history. To make this tool practical, high data linkage accuracies shall be achieved. We have developed methods towards this goal by addressing the three questions that motivated this PhD research. When combined together, these methods form a systematic solution for historical census data linkage.

From the application point of view, the systematic solution for historical census data linkage introduced in this thesis is one of the first that integrates data cleaning and standardisation, as well as record and household linkage, over consecutive census. Our solution consists of several data pre-processing, machine learning, and data mining methods that address different aspects of the historical census data linkage problem. The effectiveness of this solution has been verified on six historical census datasets collected from the United Kingdoms (UK), as will be discussed in Chapter 3. Moreover, our solution is general in nature, and can be applied to other historical census datasets if they share the common nature as the datasets used in our research.

From the technical point of view, a common innovative property shared by the methods reported in this thesis is that they all adopt a strategy to consider a household as an integrated entity, and use the whole household information to improve their effectiveness. On one hand, the household information has been used in the data cleaning and standardisation step to find errors in several key attributes, as well as in the data linkage step to enable effective reduction of ambiguity of multiple household and record linkages.

The household information has been used in different ways. In the group linking method reported in Chapter 5, individual record similarities and the number of household members are used. In the data cleaning and graph matching methods described in Chapters 4 and 6, respectively, the relationship between household members are considered. In the multiple instance learning method proposed in Chapter 7, similarities of both matched and non-matched household members contribute to the modelling process.

The contribution of this thesis also comes from the adoption of the latest development of data mining and machine learning approaches for group data linkage, which is

exemplified by group linking, graph matching, and multiple instance learning. These methods are extended both theoretically and practically to make them suitable for the group linkage task.

1.4 Thesis Outline

The remainder of the thesis is organized as follows.

Chapter 2 summarises state-of-the-art methods for data linkage. This chapter covers methods that are related to data cleaning and standardisation, string similarity comparison, and general pattern classification methods. We also give a review on historical census data linkage methods, and analyse their advantages and disadvantages.

Chapter 3 introduces the historical census data used in our research. We give detailed descriptions on when, where, what, and how the data were collected. The key features of these data are introduced, followed by an analysis on data structures and their quality, which include basic statistics on attribute values and types of errors in the data.

Chapter 4 describes historical data processing methods so as to improve the data quality and extract useful information for the following data linking step. Based on the analysis results from the previous chapter, methods to automatically clean and standardise the historical census data are described. This chapter also includes a domain driven method to automatically identify and mark households in the datasets. Such information is used to further clean the historical census data. Finally, this chapter introduces methods to compute the attribute-wise similarity of individual record pairs using approximate string similarity measures.

Chapter 5 introduces a novel method to link households in historical census data based on group linking. This method first performs pair-wise record linking using either a similarity threshold or a Support Vector Machine (SVM) classifier. This allows classification of individual record pairs into matches and non-matches. In a second step, a group linking approach is employed to link households based on the matched individual record pairs, which also increases the likehood that the correct individual link from a number of candidate links is selected. Experimental results show that this method greatly reduces the number of multiple household matches compared with a traditional linkage of individual record pairs only.

Chapter 6 explores a graph matching method for linking households between historical census datasets, which takes the structural relationship between household members into consideration. Based on individual record linking results, this method builds a graph for each household, so that the matches are determined by both attribute level and record relationship similarities. Experimental results show that such structural information is very useful in the household linkage step, and when combined with a group linking method, can generate very reliable linking outcomes.

Chapter 7 introduces a novel household linkage method based on Multiple Instance Learning (MIL). This method treats group links as bags and individual record links as instances[2]. Then multiple instance learning is extended from bag to instance classification to reconstruct bags from candidate instances. Several instance reconstruction strategies are proposed and compared. The classified bag and instance samples lead to a significant reduction in multiple group links, thereby improving the overall quality of linked data.

Finally, we summarize the thesis in Chapter 8, discusses the main results achieved

[2]Bags and instances are basic concepts in multiple instance learning, which will be introduced in Chapter 7.

in our research, and draw conclusions. We also propose future work directions in historical census data linkage research, and more general group based data linkage research.

1.5 Publications

1. **Automatic Cleaning and Linking of Historical Census Data using Household Information**
   Zhichun Fu, Peter Christen and Mac Boot.
   Proceedings of the Fifth International Workshop on Domain Driven Data Mining (DDDM'11), held at IEEE ICDM, pages 413-420, Vancouver, December 2011.

2. **A Supervised Learning and Group Linking Method for Historical Census Household Linkage**
   Zhichun Fu, Peter Christen and Mac Boot.
   Proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), pages 153-162, Ballarat, December 2011. CPRIT, 121.

3. **Multiple Instance Learning for Group Record Linkage**
   Zhichun Fu, Jun Zhou, Peter Christen and Mac Boot.
   Proceedings of the Sixteenth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'12), Kuala Lumpur, May-June 2012, pages 171-182, LNAI 7301, Springer.

4. **A Bag Reconstruction Method for Multiple Instance Classification and Group Record Linkage**
   Zhichun Fu, Jun Zhou, Furong Peng and Peter Christen.
   Proceedings of the Eighth International Conference on Advanced Data Mining and Applications (ADMA'12), pages 247-259, Nanjing, China, December 2012.

5. **A Graph Matching Method for Historical Census Household Linkage**
   Zhichun Fu, Peter Christen and Jun Zhou.

Proceedings of the Eighteenth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'14), Taiwan, May 2014.

6. **Automatic record linkage of individuals and households in historical census data**

   Zhichun Fu, Mac Boot, Peter Christen and Jun Zhou.

   The International Journal of Humanities and Arts Computing, October 2014.

Chapter 2

Background

2.1 Introduction

Record linkage, also known as entity resolution [11, 12], object identification [13, 14], or data matching [15], is the process to identify the same entity from one or more electronic files. This technique has been widely used as a pre-processing step for generating high quality data before data mining, and in many real-world applications.

In this chapter, we first give a brief introduction of the history of record linkage. We then describe the procedure of record linkage, which includes several important steps such as data cleaning and standardisation, string similarity comparison, and classification. For each step, we review some classic methods that are relevant to the methods developed in this thesis. Finally, a review on historical census data linkage methods is given, followed by an analysis on their advantages and disadvantages.

2.2 Brief History of Record Linkage

The initial idea of record linkage can be traced back to 1946, when Halbert L. Dunn introduced this term to link health care data for personal life file creation [16]. Dunn wrote: "Each person in the world creates a book of life. The book starts with birth and ends with death. Its pages are made up of the principle events of life. Record linkage is the name given to the process of assembling the pages of the book into

a volume." He also mentioned that accuracy of some vita records would be greatly improved through the linkage step.

In 1959, computerised record linkage was first carried out by a Canadian geneticist Howard Borden Newcombe and his colleagues [17], who proposed that valuable information could be acquired by bringing records from different sources together. Departing from this idea, their research was focused on developing methods to find unique identities from multiple data sources. When doing so, the difficulty came from the unreliable and ambiguous information contained in household records. In order to address uncertainty in the data, a phonetic coding approach, i.e., Soundex code [18], was applied to names in both birth and marriage records which were to be linked. This allowed the reduction of the number of record links to be processed in later steps. Then the frequency of values in different attributes were used to calculate the probability that two records are matched or not. In a later paper, Newcombe et al. commented that such statistics shall be drawn from large collections of linked record pairs in order to achieve the maximum accuracy of linking outcomes [19].

Fellegi and Sunter followed Newcombe's ideas to develop a mathematical model which provided a theoretical framework for a computer-oriented solution to the problem of recognising records in two files which represent identical persons, objects, or events [20]. In the model by Fellegi and Sunter, a linkage rule is that pairs of records between two databases in a product space can be classified as matched, unmatched, and possible matched. A decision rule is that if the matching score of a record pair is larger than an upper threshold, the pair is designated as a true match. If the matching score is smaller than a lower score, the pair is a designated non-match. Otherwise, it is a designated possible match and the decision is subject to manual clerical review.

Fellegi and Sunter's theory was improved by William Winkler [21] in considering and modelling the statistical relationship between attributes. Different from early works which assumed that attributes are independent from each other, Winkler modeled their conditional dependencies, and used the Expectation-maximization (EM) algorithm to estimate the latent matching variables [22]. He showed that such matching parameters vary greatly from dataset to dataset, for example, on data captured from different geographic regions. Therefore, the match and non-match probabilities estimated under attribute dependence are more accurate and can adapt to various datasets for practical usage.

During the past twenty years, research on record linkage has been promoted by researchers working on real-world applications. An impelling need comes from queries and merging of patient/client records in clinical data repositories [23, 24, 25, 26, 27]. A number of approaches have been developed, which can be roughly divided into deterministic or probabilistic. The deterministic approaches aims at finding exact agreement on all or a predefined subset of linking attributes in order to determine the matches [28, 29, 30]. Probabilistic linkage, on the other hand, follows the rationale of Fellegi and Sunter in calculating scores for matching or non-matching between records based on the probability estimated from attributes [31, 32, 33, 34]. Tromp et al.[35] compared the deterministic and probabilistic approaches on simulated data, and found that the full deterministic approaches can produce the lowest false positive links, but at the cost of missing large numbers of true matches. The probabilistic approaches had outperformed the deterministic strategy across all scenarios because it can tolerate a disagreement between attributes [35]. Sometimes, both deterministic and probabilistic approaches are used. For example, Schraagen directly accepted the exact matches when linking birth, marriage, and death certificates in the Dutch historical civil certificates database. The possible matches are accepted if they meet some pre-defined criteria [36].

Real-world applications that require record linkage also include bibliographic record linkage [37, 38, 39], commercial customer record linkage [40, 41], criminal identification [42], genealogical record linkage [43], and historical census data linkage [44] which is the problem targeted in this thesis. Large-scale databases are generated out of these applications. Examples include bibliography databases such as PubMed[1], ACM Digital Library[2], IEEE Xplore[3], and DBLP[4]. Other examples include public health record datasets such as the Utah Population Database[5], Washington State's Division of Alcohol and Substance Abuse [28], and Shared Health Research Information Network (SHRINE) [45]. The research support also comes from the development of data processing and linkage toolboxes, such as Febrl [46, 47, 48], FRIL [49], and D-Dupe [50].

In more recent years, thanks to the rapid development of machine learning research, various learning approaches have been introduced into record linkage methods to improve the accuracy of record linkage [51, 52]. Among them, rule-based approaches have been widely used [53, 54]. The decision of rules normally follows strong domain knowledge of the data to be linked [55]. Furthermore, other learning models such as unsupervised learning [56], supervised learning [10], online learning [57], and active learning [58] have all been explored. Some of the approaches that are relevant to this thesis will be introduced in later sections. More detailed literature reviews can be found in several survey papers [59, 60, 61].

2.3 Record Linkage Process

The traditional record linkage method consists of four steps, which are illustrated in Figure 2.1. The first step is data cleaning and standardisation, which aims at improv-

[1] www.ncbi.nlm.nih.gov/pubmed
[2] http://dl.acm.org
[3] ieeexplore.ieee.org
[4] dblp.uni-trier.de/db
[5] http://healthcare.utah.edu/huntsmancancerinstitute/research/updb

ing data quality. The second step is blocking, whose goal is to reduce the number of record pairs for further processing. The third step is record comparison, which is normally implemented by measuring the similarities or distances between two records. Many similary and distance metrics have been used for this purpose. The fourth step is record pair classification. As pointed out by Fellegi and Sunter [20], candidate record pairs can be classified as matched (two records belong to the same entity), unmatched (two records do not belong to the same entity) and possibly matched, based on the output of the third step. Then the quality of record linkage can be evaluated by a human operator, with special focus on further investigating those possible matches. On the other hand, qualitative evaluation measures, such as accuracy, precision, and recall, which are to be defined in Chapter 5, have also been widely used. More detailed discussion on each step will be presented in the following sections.

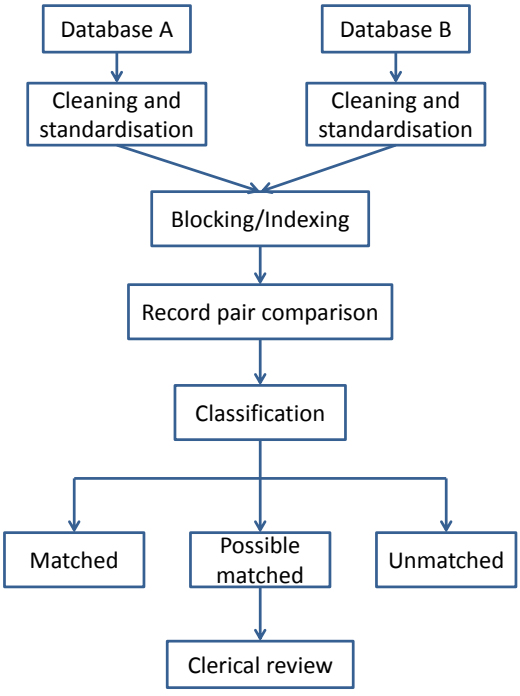


Figure 2.1: Record Linkage Process.

2.3.1 Data Cleaning

Low quality data is a problem generated during the data collection stage [62]. Data can contain missing values, wrong values, and inconsistent values. Record linkage on such data may produce wrong results, and in turn, propagate errors into the following data analysis steps. To solve this problem, data cleaning and standardisation are the key steps. They aim at improving data quality and increasing the likelihood of finding true matches within and between datasets.

Data cleaning can be divided into two steps. The first step aims at detecting low quality data. This is achieved via data analysis, which identifies locations and types of errors. The second step is to fill in missing values, correct identified errors, and remove data inconsistencies [41]. This step is closely related to the actual record linkage, in which a common practice is to use the relationship between records to fill and verify values, while using a lookup table to correct wrong values [63, 64].

Data standardization is the step of converting data stored in various data sources with different formats to a comparable structure. The problem of incompatible data format often happens during the data collection process, when data are collected in different time periods or by different people. For example, the name "John Smith" can be expressed as "Smith, John" in one dataset, but might be written as "J. Smith" in another dataset. Such incompatible formats have to be unified before further processing. The basic standardisation target is spelling [64]. Spelling standardisation converts different representations of the same word into one designated spelling. It is mostly used on names and addresses, for example, "Doctor" and "Dr." is replaced by "DR", "Bill" is replaced by "William", and "St." is replaced by "Street". Most spelling standardisation is based on lookup tables or dictionaries, and sometimes combines spelling rules with manual checking [44, 65]. Consistency of coding is also part of the standardisation. It converts attribute values to the same coding scheme, for example,

converting all dates into the YYYYMMDD format. Another example is that an age value “under 1" (years old) can be converted to “0" (years old), i.e., numerical year format.

Limited research has been conducted in data standardisation. For example, a rule-based method has been used in the date standardisation process by Christen et al. [66, 67]. The standardisation of names and addresses in this method was based on hidden Markov models (HMMs). The authors claimed that HMMs are simpler and less time consuming than other traditional rule-based methods. Their results showed that the reported method works well for address standardisation but poor for name standardisation compared with rule-based methods. Morillo et al., compared hand-coded address data with those generated by a semi-automatic method which requires certain amounts of human input [68]. Each research institution in the Web of Science database was assigned a unique address to form a reference list, then all other addresses were standardized against this list. The results showed high agreement between the manual and semi-automatics methods, and therefore, validated the effectiveness of the semi-automatic method. In [69], a supervised learning method was proposed to ease the human efforts of rewriting rules for annotating large amounts of data. The idea was to capture the latent semantic association among words from unlabeled data, and capture the data distribution of the target domain for address standardization.

Some software packages have been developed for data cleaning and standardisation purposes. For example, Febrl [46, 47, 48] provides data cleaning, data standardisation, record deduplication and linkage functions, by implementing state-of-the-arts methods using Python. This software uses Australian national address guidelines and other national address databases to build inference structures, and has achieved high accuracy for complex and unusual addresses. FRIL [49] also provides pre-

processing functions such as standardisation, as well as splitting and merging of attribute values.

2.3.2 Blocking

With the development of information technology, the speed with which data are generated and collected has increased dramatically. Larger databases make the record linkage process a challenging task. Suppose two datasets have to be linked, with X and Y records in each dataset respectively. An exhaustive comparison of all record pairs will take up to $X * Y$ comparisons. When X and Y are very large, the traditional linkage approaches become infeasible.

The development of blocking techniques aims at solving this problem. It speeds up the linking process by subdividing data into several blocks using blocking keys. Here, the assumption is that true matches only happen within the same blocks. Then, detailed comparisons between records are executed only between the records that have the identical blocking key. For example, the first three letters in a first name attribute can be used as a blocking key. All names with the same blocking key value will be put into one block. This significantly reduces the number of comparisons to be performed in the following similarity calculation and matching steps.

In recent years, various blocking methods have been developed, including traditional blocking [70], sorted neighbourhood [71], Q-Gram based blocking [72], Canopy Clustering [73], and Suffix Array blocking [74]. Traditional blocking only compares the records in a block that have an identical blocking key value, which are usually chosen from one or more attributes from each dataset. The selection of blocking key(s) should be done very carefully so that many matched records can be included into the same block. A normal practice is to select attributes with no or only small amounts of errors and missing values, and with average frequency distribution.

The sorted neighborhood method sorts records based on a sorting key. Then a sliding window of fixed size is moved over the sorted records sequentially [71]. Comparison is limited to the records within the window. Although this method can reduce the total number of comparisons, there are two main disadvantages. Firstly, when the number of records with the same sorting key is larger than the designated window size, not all potentially matched pairs will be compared. Secondly, this method is not tolerant to errors at the beginning of attribute values. For example, when the first letter of the sorting key contains an error, similar records will be assigned to different blocks, and, as a consequence, they will not be compared with each other. Lehti [75] modified the original sorted neighborhood method in [71] by replacing the fixed size window with a dynamically sized window based on distance measured between the keys. The evaluations showed that such a modification can significantly improve linkage accuracy over the original method while keeping the computational costs low.

Q-Gram based blocking is to convert blocking key values into a list of substrings based on the parameter Q. Q is the length of a substring [72]. The sublists of all possible permutations of Q-Grams are built using a threshold. The resulting lists are sorted and the corresponding records are retrieved in a block. The computational complexity of Q-Gram based blocking is dependent upon the parameter Q and the threshold value. Thus, the main drawback is that a small Q value and low threshold will generate a large number of short sublists, which makes the comparison become very time consuming. Research was shown that the Q-Grams method can achieve better results than traditional blocking methods and sorted neighborhood methods [72].

Canopy Clustering [73] creates blocks of records by inserting records into the same canopy cluster, while canopies can be overlapping. The canopy clusters are created

by calculating the similarities between blocking key values using measures such as Jaccard or TF-IDF/cosine similarity [41]. In [73], the authors showed that the canopy clustering method can reduce the computational costs by more than an order of magnitude over the traditional blocking method, while reducing errors by 25%.

Aizawa and Oyama proposed a Suffix Array based blocking method to insert blocking key values and their variable length suffixes into a suffix array structure [74]. Then blocks are generated based on these suffix arrays. This method can better overcome errors at different locations in the blocking key values because a record will be inserted into several blocks based on the new blocking key established. De Vries et al. improved the suffix array blocking method by merging similar inverted index lists of suffix values into the sorted suffix array [76]. A similarity measure is used to calculate all pairs of neighboring suffix values. If the measured similarity of a pair is higher than a threshold, two lists are merged into one new block [41].

Machine learning approaches have been used to automatically optimise blocking methods. In [77], a blocking method is introduced to learn a blocking scheme using a modified Sequential Covering Algorithm (SCA). SCA learns one conjunction of attributes, removes the training records it covers, then iterates the process until reduction ratio, which is the number of records to be removed out of all remaining records, converges. The authors compared their blocking schemes with four ad-hoc blocking schema learning algorithms. The results show that the proposed method outperforms the non-experts alternatives and performs comparably to those manually built by a domain expert. Whang et al. proposed an iterative blocking framework [78]. The results of entity resolution after blocking are used to generate new record matches in other blocks, which initiates another round of blocking. Such iteration continues until no blocks contain known matches. The approach has achieved high efficiency and more accurate results compared to other blocking methods.

Several encoding techniques have been used in the blocking step to transfer a string into a special code that will bring together similar strings into one block. Two common encoding methods are Soundex and Double Metaphone [61]. Soundex is a classical phonetic algorithm. As the name indicates, the Soundex algorithm aims at matching strings that “sound" similar. The algorithm keeps the first letter of a string unchanged and converts the remains letters to a new code based on an encoding table. For example, “licence", “license" and "licensing" are mapped to the same Soundex string “L252". The Double Metaphone algorithm improves some encoding choices made in the initial Metaphone algorithm [79]. It allows multiple encodings for strings that have various possible pronunciations. This implies checking all possible encodings for similar strings retrieval. For example, the Double Metaphone codes for “Schneider" are XNTR and SNTR. As mentioned in [60], 10% of American surnames have multiple encodings. Thus, the Double Metaphone algorithm can greatly enhance the matching performance.

2.3.3 String Comparison

Traditionally, deterministic or exact matching rules are used for record linkage. Under these rules, if attributes in two tuples have the same value, these two records are considered to be referring to the same entity and can be matched. Such rigorous matching criteria cannot deal with most real world data, which contain typographical errors, acronyms, and missing values. Winkler has pointed out that more than 25% of matches could not be found using exact matching in major census applications [59]. Therefore, accurate exact matching can only be obtained under the assumption that data are free of errors, that is, the zero error tolerance assumption.

Since typographical error is inevitable in almost all real world data, different methods have been studied to find approximate matches between strings. These methods can be classified into two types [80]. The first type is phonetic encoding methods,

such as Soundex and Double Metaphone, which have already been introduced in the previous sub-section. The second type is pattern matching methods, such as edit distance and q-grams. In record linkage, approximate string comparison similarity results are normalised into $[0, 1]$. The higher a value, the more similar two strings are. Thus, a 1 indicates an exact match and 0 means no similarity.

Edit distance computes the minimum number of edit operations that are necessary to convert one string into another. This is done by measuring common typing errors, such as character insertions, deletions, and substitutions [41]. It is easy to prove that Edit distance is a proper distance metric, therefore, the conversion between similarity and edit distance of two strings $s1$ and $s2$ can be calculated as following:

$$sim(s1, s2) = 1 - \frac{ed(s1, s2)}{max(|s1|, |s2|)}, \tag{2.1}$$

where $sim(s1, s2)$ is the similarity between $s1$ and $s2$, $ed(s1, s2)$ is the edit distance, and $|*|$ is the length of a string.

The Jaro similarity metric [70] was first introduced in 1989. It takes into account typical spelling variations, including insertions, deletions and transpositions. For two strings, the Jaro similarity metric calculates the length of each string, the number of common characters in the two strings, and the number of transpositions. It is given by

$$Jaro(s1, s2) = \frac{1}{3}(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}) \tag{2.2}$$

where $s1$ and $s2$ are two strings, m is the number of common characters in these strings, $|s1|$ is the length of the first string, $|s2|$ is the length of the second string, and t is the number of transpositions (number of non-matching characters). The characters are considered matching only if their distance is less than or equal to:

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1. \tag{2.3}$$

Equation 2.3 suggests that this method works best on strings of similar length. When the lengths of two strings differs too much, the distance of matched characters may be larger than the threshold defined in Eq 2.3, and thus, matching characters cannot be found.

Winkler extended this method by introducing an extended similarity measure [21]. This measure uses a prefix scaling to modify the weights of string pairs, which gives more favourable ratings to strings that match from the beginning for a set prefix length. The Jaro-Winkler similarity metric is defined as

$$Jaro_{winkler}(s1, s2) = Jaro(s1, s2) + \ell p(1 - Jaro(s1, s2)) \tag{2.4}$$

where ℓ is the length of common prefix at the start of the strings, $\ell \leq 4$, and p is a constant scaling factor that defines the extent that the score is adjusted upwards for having a common prefix, with restriction that $p \leq 0.25$. In Winkler's work, $p = 0.1$.

Q-Grams, which were introduced in the previous section, have shown to be one of the most simple and efficient approaches for pattern matching besides edit distance. They were originally developed by Ullmann in 1977 [81], and later extended to commercial relational databases by Gravano et al. [82]. In [82], a q-gram algorithm is combined with relational schemes so that it is not necessary to change the underlying database system. Three different methods can be used to compare two strings based on their q-grams. They are overlap coefficient, Jaccard similarity, and Dice coefficient [80]. All these methods return the total number of common q-grams divided by the number of q-grams in the short string, the number of long strings, or the average number of both strings. Given two strings s and t. Let S and T denote the q-grams sets for s and t, $|s|$ and $|t|$ denote the length of strings s and t, and $|S|$ and $|T|$ denote the number of q-grams in s and t based on the value of q, respectively. Then, $|S|$ can be calculated as $|S| = |s| - q + 1$. The overlap coefficient, Jaccard similarity,

and Dice coefficient are defined as:

$$overlap(s,t) = \frac{|S \bigcap T|}{\min(|S|,|T|)} \tag{2.5}$$

$$Jaccard(s,t) = \frac{|S \bigcap T|}{|S \bigcup T|} \tag{2.6}$$

$$Dice(s,t) = 2 * \frac{|S \bigcap T|}{|S| + |T|} \tag{2.7}$$

The difference of these three comparison methods can be illustrated by the following example. Let $q = 2$, the q-gram becomes a *bigram*. The *bigram* can obtained by sliding a window of length 2 over a string. For instance, “Tim" contains the *bigram* set $S = \{ti, im\}$ and “Timothy" contains the *bigram* set $T = \{ti, im, mo, ot, th, hy\}$, for which we can get $|S| = 2$, $|T| = 6$. The similarity calculated for the string comparison methods are $overlap(s,t) = 1$, $Jaccard(s,t) = 1/3$ and $Dice(s,t) = 1/2$.

In [83], Porter and Winkler examined various string similarity comparators on a modern probabilistic record linkage system. They concluded that all of these string comparators can greatly improve the matching results over the exact matching method.

2.3.4 Classification

Many machine learning methods have been used in record linkage to classify pairs of records into the match, unmatch, and possible match classes. Due to the complex relationship between records in a dataset or across several datasets, group based classification methods have also been introduced into record linkage. In general, machine learning approaches can be divided into supervised or unsupervised techniques. When a set of labeled data is used to train a classifier, it is called a supervised learning method [84]. However, labeled data are normally expensive and the labelling process is often time consuming, especially on large datasets. As a consequence,

most likely there are no labelled data available at all, which requires unsupervised learning solutions.

2.3.4.1 Record Pair Classification

Supervised learning uses a training set (labeled examples) to learn a classification model, and then applies the model to testing sets (unlabeled examples) in order to predict the classes of each data sample [51, 85, 86, 87, 88]. During the past decades, the machine learning and pattern recognition communities have developed various supervised learning approaches. Among them, decision trees [89, 90], logistic regression [91], neural networks [92, 93], support vector machines [94, 95], and Bayesian classifiers [22] have all been applied to record linkage.

In our work, a Support Vector Machine (SVM) classifier has been used in several record linkage models to provide either record level or household level classification. SVM was developed by Vapnik [96]. It aims at computing a hyperplane(s) to classify data mapped into a high dimensional space via a kernel function. A key point here is to construct the kernel matrix for which an SVM can be used to perform the training and classification. Bilenko and Mooney proposed such a solution [94] to compute the similarities of strings as kernel matrix directly. Similarly, Nahm et al. [97] used an SVM to classify record pairs after attribute-wise similarities calculation. Alternatively, Christen [95] constructed inputs to an SVM using a pre-selection step. The threshold selection or nearest-based selection were used to select record pairs with high confidence of being matching and non-matching. Then these pairs became the positive and negative training samples for the SVM classifier. This method can be considered as a combination of supervised and unsupervised methods.

Unsupervised learning, such as clustering, does not reply on class-labeled training examples. The most popular clustering algorithm is the k-means clustering [63]. It

adopts an iterative updating methods to partition a set of data samples into k clusters in order to minimize the within class distance. Elfeky et al. developed an interactive record linkage tool box, Tailor [90], in which the k-means clustering was used as one of three machine learning approaches to assign candidate record pairs into a matched, unmatched, and possibly matched clusters. Collective entity resolution (or collective linkage) techniques [11] use information that explicitly connects records to collectively compute all links between records from two datasets in an overall optimal fashion. Experimental studies (mostly on bibliographic data) have shown that these techniques can improve linkage quality significantly compared to the traditional approaches that only consider pair-wise similarities between individual records.

Another unsupervised learning method is relational clustering [11], introduced by Bhattacharya et al., which links related entities between the co-occurring references and combines them with the attribute similarity measures. They investigate the effectiveness of these relational similarity measures on three real bibliographic datasets and synthetically generated data. The method shows better performance compared with attribute based algorithms and a naive relational algorithm. Similar to [11], Kalashnikov and Mehrotra [98] combined traditional feature-based similarity (FBS) algorithm with inter-object relationships to improve disambiguation quality. They also used graph theory to discover and analyze relationships between the references and the set of candidates. Nuray-Turan et al. extended the work in [98] and built a graph model and labelled dataset to compute the strength of connections among linked candidate records [99]. A self-tuning approach is developed to update the model in a linear programming fashion.

2.3.4.2 Group-based Classification

In many cases, we need to consider the relationship between a group of records instead of pairs of records only. This is sometimes because multiple records from

different data sources have to be linked [100, 101], for example linking the same article across several bibliographical databases [102], or linking homicide data provided by different resources [9]. In other cases, the relationship among multiple records will be used to determine matching or non-matching of record pairs, for example, linking households across census datasets collected at different times [44] (as will be discussed in details in Chapter 6).

To address such scenarios, group record linkage methods have been developed to process groups rather than individual records [102]. On et al. [103] defined group similarity from two aspects, either the similarity between matched record pairs or the fraction of matched record pairs between two groups of records. A group similarity can then be calculated using a maximum weight bipartite matching. To further model the relationship of more than two records in multiple datasets, Sadinle et al. [9] extended the traditional Fellegi and Sunter [20] model to calculating the joint probability of multiple records. A decision rule is proposed under the condition that true matching probabilities are available. Further modelling of the relationship between multiple records in a dataset or among several datasets can be done by exploring the latest machine learning approaches that deal with groups of data, in particular, graph-based models and multiple instance learning methods.

Graph-based Learning

A graph-based approach is a natural solution to model the structural relationship between groups of data. During the past years, several graph matching methods have been proposed to match records. Domingos proposed a multi-relational record linage method to de-duplicate records [37]. This method defines conditional random fields, which are undirected graphical models, on all candidate record pairs. Then a chain of inference is developed to propagation matching information among linked records. Hall and Fienberg reported a method to build bipartite graphs and evaluate

the confidence of different hypothetical record link assignments [104]. This method can be used to link datasets of moderate size. Furthermore, hierarchical graphical models have been proposed to cope with the potential structure in large amount of unlabeled data [105].

Multiple Instance Learning

Multiple instance learning is a paradigm of machine learning that deals with a collection of data called *bags*. The data samples in a bag are called instances. Therefore, a bag may contain a number of instances. The class label is only available at the bag level. A positive bag contains both positive and negative instances, and a negative bag contains negative instance only. The original work by Dietterich et al. [106] attempted to recover an optimal axis-parallel hyper-rectangle in the instance feature space to separate instances in positive bags from those in negative bags. Departing from this model, several researchers have extended the framework, such as MI-SVM [107], DD-SVM [108], SMILE [109], MILES [110] and MILIS [111]. In the historical census data linkage, a household link can be considered as a bag, and all corresponding record links are the instances. Then we can train multiple instance models to predict whether household and record links are matched or not. The details will be discussed in Chapter 7.

Among these works, we are particularly interested in the Multiple Instance Learning with Instance Selection (MILIS) method because it allows efficient and effective instance prototype selection for target concept representation [111]. MILIS is an extension of MIL using an embedded instance selection (MILES) method [110]. The general idea of these two methods is to map each bag into a feature space defined by the selected instances, which is based on bag-to-instance similarity. They generates instance prototypes from training bags, then the similarities between a bag and these instance prototypes can be calculated using a Hausdorff distance. This allows

the embedding of bags into a vectorised feature space, so that an MIL problem is converted into a supervised learning problem, for which a traditional supervised learning approach, such as SVM, can be used to learn a bag model and use it to predict the label of new bags. The difference between MILES and MILIS is that MILES uses all instances in the training bags for embedding but MILIS selects only one instance from each bag for the same purpose. Therefore, MILIS supports a much lower dimensional feature space than MILES, which allows efficient data processing. This is an important property for (historical) census record linkage, a problem targeted by this thesis, which works on potentially large numbers of households and their records, and contains significant amounts of uncertainty because of low data quality. The details will be discussed in Chapter 7

The major difference between the MILES and MILIS methods is on the instance selection step. In MILES, all instances in the training set are used for feature mapping, then important features are selected by a 1-norm SVM. Because the total number of instances in a training set may be very large, MILES can be very time consuming. MILIS, however, only selects one instance prototype (IP) from each bag for the embedding. It generates a feature space with much smaller dimension than MILES. The selection of IPs is done through a two-step optimisation framework, which updates IPs and an SVM classifier iteratively.

2.4 Historical Census Data Linkage Methods

Social scientists have linked census records for decades using both manual and automatic methods. Schürer from the University of Essex has led a team to process and link computerised nineteenth-century census collection in the United Kingdom [112, 113]. Their work include data cleaning and digitization, patterns of employment and occupation structure, and geographic name distribution. These preprocessing and analysis steps are crucial in creating a usable historical census col-

lection, and then extending it for social science research. Larsen investigated the problem from a probabilistic point of view using a maximum likelihood estimation model to separate record pairs into possible matches and non-matches [114]. Manual checking was then performed to update the estimation model. This process was iterated until few additional matches remained. Ruggles attempted to limit false matches by selecting attributes that did not change over time. Ambiguous links were removed to achieve high rates of linkage accuracy [3]. The effect of this method is that a large number of links including many correct links may be missed due to the removal. Vick et al. standardised name strings in a population study of census data from the United States and Norway [47, 115]. The authors used name dictionaries and estimates of name frequencies to select how name values were to be cleaned and standardised. The Jaro-Winkler approximate string comparison algorithm was then used to match candidate names to their standard form [59]. The effectiveness of the standardisation was validated in that it greatly reduced the number of false links.

The attributes used in these methods vary greatly depending on the detailed personal information that is available in the census dataset. The most commonly used attributes include first and last names, house number, street name, phone number, age, birth year, birth place, parent's birth place, relation to head of household, marital status, sex, and race. The sizes of the datasets used in these approaches vary a lot. For example, Larsen and Rubin validated their method on five US Census/Post-Enumeration Survey datasets with more than 288,414 links containing 26,315 matches [114]. Ruggles studies 500,000 individuals in the Church of Jesus Christ of Latter-Day Saints database, and one per cent of the Integrated Public Use Microdata Series [3].

Existing efforts to improve linkage accuracy are mostly focused on standardising names, removing ambiguous links, or combining different attributes to improve sim-

ilarity scores for the compared record pairs and to deal with low data quality problems (discussed below) that have seriously hampered the improvement of linking accuracy in previous works. For example, name standardisation methods developed by Vick and Huynh have increased the number of single matches but have failed to reduce the problem of multiple ambiguous links [115]. The result is that many methods simply discard ambiguous links, which leads to the loss of large amounts of potentially true matches, or have left these for manual checking [3, 114, 116].

Goeken et al. attempted to deal with the inaccuracy of 19th century census data by generating initial linkage results using name and age similarity scores, name commonality, and birthplace distribution measure [116]. They then use the single record links that have a very high confidence value as primary links to identify matched households. Once matched, the linked households then allow them to assume that other members resident in both households are also confirmed positive matches even though they might have low similarities that would otherwise have been treated as ambiguous and therefore be rejected. The distinct feature of their approach, however, is that even though they use household information, their linking step remains dependent on the initial single record link.

Antonie et al. reported a complete system to link people in multiple census collections from 19th century Canada [117, 118]. Their system consists of several key steps, including data cleaning, string comparison and processing, feature construction, blocking and thresholding, and record pair classification. The feature construction step uses several combinations of attributes and distance measures. The name attribute comparison uses edit distance, the Jaro-Winkler similarity function, and Double Metaphone, which leads to eight name features. Age comparison generates a binary code on whether the ages of two people match or not. The gender, birth place, and marital status comparison is based on exact match. The concatenation of these

features forms a feature vector for a pair of records, which is classified by a support vector machine. There are two problems with this classification system. First, if a record is matched to multiple records, it is simply discarded. This will cause the missing of many true matches. Second, this linking system is still record based, and has not taken household information into consideration.

Chapter 3

Historical Census Data and Basic Analysis

In this chapter, a brief introduction is given on the historical census data that have been used in our research. This is followed by data analysis which helps to obtain a deeper understanding of the structure, statistics, data quality, and the challenges of historical census data linkage.

3.1 Introduction

A census is a complete statistical count that records people who live in a country[1]. It is one of the most complete and important tools that governments use for policy and decision making, such as estimating the population of an area, planing resource and funding distribution, and city planning. Social scientists have also been using census data to reconstruct various aspects of societies of the past and the present.

In our research, we use historical census data collected from the United Kingdom. The earliest national census in the UK was taken place in 1801 [1, 119]. Since then, a census was taken every ten years except 1941 due to World War II. The earliest census with person's name is the 1841 census return. UK historical census returns were

[1] http://familysearch.org/learn/wiki/en/England_Census

collected on Census Night. Questions were put in hand-filled census forms, called schedules. The distribution of these schedules was based on small districts, which guaranteed that the schedules can be delivered and collected timely. Census enumerators delivered the schedules to each household. The householder were required to complete the form following instructions. Then the enumerators were instructed to check the schedules on the doorstep when they collected them. Enumerators then copied them into census enumerators' books. Public access to census returns in the UK are under the 100 years census disclosure policy[2]. This is the reason that the most common UK census data used for research are from 1841 to 1911[3].

3.2 The Rawtenstall, Lancashire Censuses of 1851 to 1901

Figure 3.1: Map of England. The red area is Lancashire, from where the census data used in our research was collected.

In our research, we use six historical census datasets covering the district of Rawtenstall in North-East Lancashire in the UK (Lancashire is the red area in Figure 3.1) between 1851 and 1901. The data was provided by "The Rawtenstall Project" of the Australian Demographic and Social Reseach Institute in the Australian National Uni-

[2](Lord Chancellor's Instrument no.12 in S.5 (1) of the Public Records Act 1958. Issued in 1966.
[3]When we started this research in 2010, we were only able to access the 1851 to 1901 data.

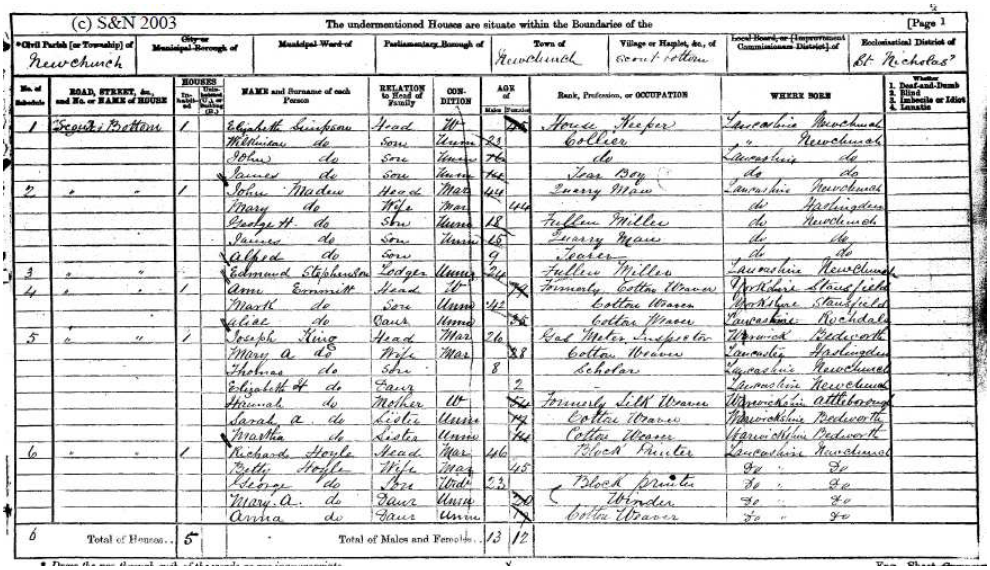


Figure 3.2: Historical census form in good quality.

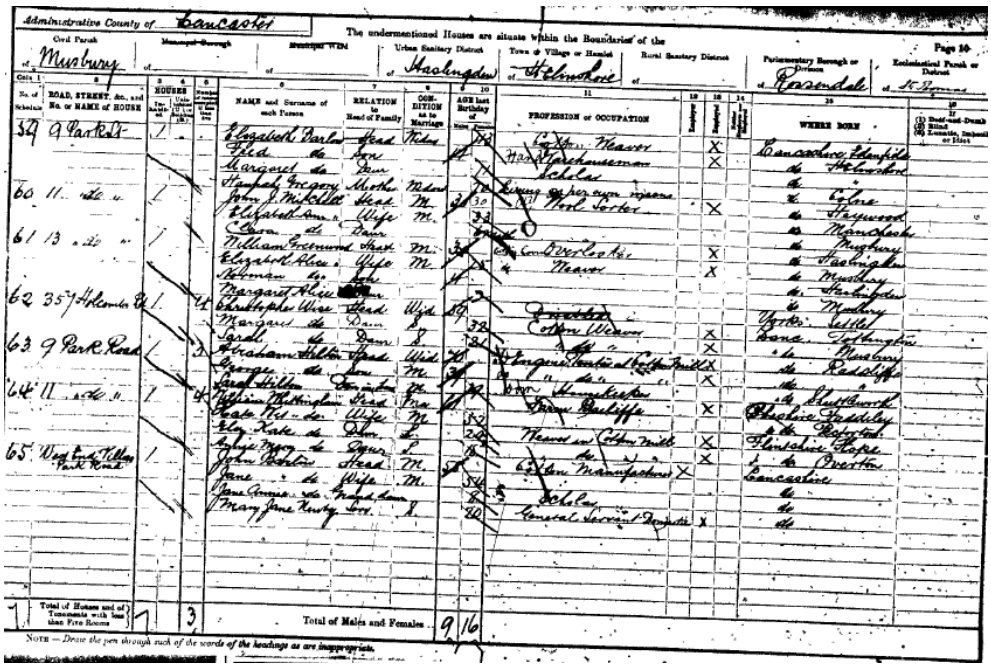


Figure 3.3: Historical census form in bad quality.

versity[4]. The original hand-written-returns have been scanned into digital form. The quality of these digital forms varies a lot, due to the way the returns were completed and scanned. Figures 3.2 and 3.3 show two samples of original images with good and bad quality. The quality problem will be propagated to later steps of data digitisation and analysis. Further analysis on the quality problem will be reported in Section 3.2.1. The last step of digitisation was a manual transcription of the digital form into tables and storing them in electronic spreadsheets. Figure 3.4 shows a sample of census data in a spreadsheet.

Image ref.	Address	Census parish	County	Surname	First name	Relation to H of HseHld	Sex	Age	occupation	Birth Parish	Birth County
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FIELDING	JAMES	HEAD	m	32	MASTER ROPES MAKER	Middleton	YORKSHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FIELDING	MARGARET	WIFE	f	30	HOUSE KEEPER	Bury	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FIELDING	MARY ALICE	DAUGHTER	f	7	SCHOLAR	Newchurch Clough Fold	-
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FIELDING	ANN	DAUGHTER	f	2	-	Newchurch Clough Fold	-
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FIELDING	SARAH ELLEN	DAUGHTER	f	4 M	-	Newchurch Clough Fold	-
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	HART	JAMES	LODGER	m	13	WINDER	Wigger	-
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FARAR	SARAH	HEAD	f	52	HOUSE KEEPER	Spotland	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FARAR	JOHN	SON	m	28	BAKER	Bury	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FARAR	ALICE	DAUGHTER IN LAW	f	30	BAKER WIFE	Bury	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	FARAR	SARAH JANE	GRAND DAUGHTER	f	10 M	-	Newchurch	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	TAYLOR	HENRY	HEAD	m	40	COTTON MANGER	Newchurch	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	TAYLOR	ELIZABETH ANN	WIFE	f	28	HOUSE KEEPER	Newchurch	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	BUTHWORTH	ALICE ANN	DAUGHTER	f	8	SCHOLAR	Newchurch	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	TAYLOR	JOHN	SON	m	2	-	Newchurch	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	SAGER	ANN	HEAD	f	42	HOUSE KEEPER	Bacup	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	SAGER	THOMAS	SON	m	14	-	Stackslead	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	SAGER	WILLIAM	SON	m	11	-	Stackslead	LANCASHIRE
RG9_3055 p.7	Wales Bank	Newchurch	Lancashire	SAGER	JOSEPH	SON	m	10	-	Newchurch	LANCASHIRE

Figure 3.4: Electronic data sample.

3.2.1 Data Analysis

From a first glance at the images and spreadsheets, we can see that census data are based on households. The information of people who lived in one household is given as a consecutive set of records(rows). Each row contains the collected information of one person, and each column expresses one attribute. These information include name, age, address, marital condition, relationship to the head of the household, occupation, and birthplace.

Some data linkage challenges and quality problems can be seen from the sample images and spreadsheets. First, the census return was filled in by hand. This makes the

[4]Contact person: Mac Boot, mac.boot@anu.edu.au

contents difficult to be recognised not only by census enumerators when they tried to put the information into enumerators' book, but also for clerks who transcribed the information into tables. Errors were introduced in these stages[5]. Second, there are sometimes marks and corrections written over key information. These marks were added by enumerators or clerks for calculating the number of data entries they had extracted, as measured by score marks they went though the items in the enumerator returns. These marks and corrections were in various coloured inks and pencils that differentiated them from the written text. Later, when the census returns were digitised into grayscale images, these colours were lost, causing the marks to obscure many data entries. Thus, the digitisation results obscured these data entries to be read correctly, and therefore more errors appeared. Third, often some information is missing. This is either because questions were left blank when the head of household completed a form, or the poor handwriting and obscured data prevented accurate transcription into computer spreadsheets.

Reading through the six historical census datasets, we find that the attributes contained in each dataset are consistent. This implies that the questions asked on the schedules were very likely the same. Furthermore, Table 3.1 shows that the dates in the year that the census was collected from 1851 to 1901 were also close to each other[6]. This makes reliable linking of these censuses across time possible. Table 3.1 also shows the number of person, including males and females in each census return. The population in the district of Rawtenstall increased steadily over time. In the 50 years between 1851 and 1901, the population grew rapidly by more than 14,000 because of the booming cotton industry. This is especially the case for the 10 years between 1851 and 1861, during which the population grew by 31.7%. The population growth slowed after 1881. This large increase of residents in the area has imposed great challenge to the data linkage tasks. Due to the expansion of the area, new

[5]http://www.professionalfamilyhistory.co.uk/Census-records.html
[6]http://familysearch.org/learn/wiki/en/England_Census

Census Night	Total number of person	Males	Females	Unknown
March 31st, 1851	17,033	8,498	8,533	2
April 8th, 1861	22,429	10,934	11,494	1
April 3rd, 1871	26,229	12,672	13,548	9
April 4th, 1881	29,051	13,948	15,103	0
April 6th, 1891	30,087	14,171	15,880	36
April 1st, 1901	31,059	14,596	16,437	26

Table 3.1: Number of records in the Rawtenstall historical census datasets.

streets were built. Households might have their street names and numbers changed even though they were not moving. This increased the uncertainty of address attributes. Furthermore, people tended to move more frequently for jobs, which caused more complex households to be recorded due to the large number of short term residents boarding with other families.

Each census dataset contains one record for each person in the district. There are 12 attributes for each record, which correspond to some important aspects of households. These attributes are listed and described in Table 3.2.

Attribute	Description
IMAGE REF	Location of the record in the scanned image database
ADDRESS	Address of the house
CENSUS PARISH	Parish of the address
COUNTY	County of the address
SURNAME	Surname of the person in the house
FIRST NAME	First name and middle name of the person in the house
REL HSEHLD	The relationship to the head of the household
SEX	Gender of the person
AGE	Age of the person
OCCUPATION	The occupation of the person
BIRTH PARISH	Parish where the person was born
BIRTH COUNTY	County where the person was born

Table 3.2: Census data attributes with definition.

The “IMAGE REF" attribute shows the connection of the data in the spreadsheet with the original location of the scanned image. This is quite useful information for social

scientists to locate the original records so that manual error correction can be performed. The values in the “COUNTY" attribute are all the same in the six datasets as all the data had been collected from Lancashire. The “CENSUS PARISH" attribute is a territorial unit related to church. The names of parish and number of different names changed significantly across time, so they cannot be used as a reliable attribute. As a consequence, these three attributes (IMAGE REF, COUNTY, CENSUS PARISH) are not useful for data linkage, and are excluded for consideration in the following processing steps.

To further understand the historical census data, we performed data exploration using the Freely Extensible Biomedical Record Linkage (Febrl) tool [46]. Tables 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8 show some statistics of the raw historical census data for each ten years census return from 1851 to 1901. These tables contain six columns. The first column shows the attribute name, followed by the number of unique values and missing values for each attribute. The type column shows the type of variables, i.e., letters, digits or mixed, while the minimum and the maximum values are displayed in the last two columns.

Attribute	Unique	Missing	Type	Min Value	Max Value
IMAGE REF.	915	0	mixed	ho2248_1 p.10	ho2249_2 p.99
ADDRESS	509	318	mixed	-	yewin hill
CENSUS PARISH	10	0	mixed	Cloughfold	Waterfoot
COUNTY	1	0	letters	Lancashire	Lancashire
SURNAME	1009	0	mixed	-	YOUNG
FIRST NAME	999	0	mixed	-	ZILPOH
REL HSEHLD	60	0	mixed	ADOPTED DAUGHTER	WIFE'S NIECE
SEX	3	0	letters	f	m
AGE	129	0	mixed	1	1M
OCCUPATION	2739	22	mixed	-	YENTERERS WIFE
BIRTH PARISH	1407	5	mixed	-	yovgend
BIRTH COUNTY	65	3	mixed	-	yorkshire

Table 3.3: Raw data quality analysis of 1851 census dataset.

Attribute	Unique	Missing	Type	Min Value	Max Value
IMAGE REF.	935	0	mixed	RG9_3055 p.10	RG9_3059 p.99
ADDRESS	666	0	mixed	1 Ballowtree	Wood Top
CENSUS PARISH	9	0	mixed	Cowpe Lenches, New Hall Hey & Hall Carr	Whalley
COUNTY	1	0	letters	Lancashire	Lancashire
SURNAME	1649	0	mixed	?	YOUTH
FIRST NAME	1483	2	mixed	?	ZILPHA
REL HSEHLD	53	0	mixed	ADOPTED DAUGHTER	WIFE
SEX	3	0	letters	f	w
AGE	122	0	mixed	0	not identified
OCCUPATION	2557	1	mixed	-	YEOMAN
BIRTH PARISH	2365	480	mixed	-	wrelston
BIRTH COUNTY	77	0	mixed	-	YORKSHIRE

Table 3.4: Raw data quality analysis of 1861 census dataset.

Attribute	Unique	Missing	Type	Min Value	Max Value
IMAGE REF.	1101	0	mixed	R10_4135 p.100	R10_4139 p.99
ADDRESS	1282	0	mixed	1 Albert Terace	whitewell Vale
CENSUS PARISH	9	0	mixed	Cowpe Lenches, New Hall Hey & Hall Carr	Rawtenstall
COUNTY	2	0	letters	Lancashire	NULL
SURNAME	1836	0	mixed	?	YOUNG
FIRST NAME	2152	0	mixed	?	pEGGY
REL HSEHLD	107	0	mixed	ADOPTED DAUGHTER	WIFES SON
SEX	5	4	mixed	?	not specified
AGE	135	0	mixed	1	not identified
OCCUPATION	3077	0	mixed	-	tINSMItH
BIRTH PARISH	2884	0	mixed	-	whalley
BIRTH COUNTY	85	0	mixed	-	YORKSHIRE

Table 3.5: Raw data quality analysis of 1871 census dataset.

Attribute	Unique	Missing	Type	Min Value	Max Value
IMAGE REF.	618	0	mixed	RG11/4129 f. 43	RG11/4136 f. 62
ADDRESS	2690	23	mixed	Haslingden Union Workhouse Pike L	longholme Front St 13
CENSUS PARISH	5	0	mixed	Cowpe Lench Newhall Hey & Hall	Tottington Higher End
COUNTY	1	0	letters	Lancashire	Lancashire
SURNAME	1959	0	mixed	(NK)	YOUNG
FIRST NAME	3390	0	mixed	...	Zipporah
REL HSEHLD	71	0	mixed	ADOPTED	WIFE
SEX	4	0	letters	F	m
AGE	132	0	mixed	1	not identified
OCCUPATION	3900	6586	mixed	((D... W...er))	winder
BIRTH PARISH	2310	2031	mixed	(British Subjec	not identified
BIRTH COUNTY	229	784	mixed	(British Subjec	scotland

Table 3.6: Raw data quality analysis of 1881 census dataset.

Attribute	Unique	Missing	Type	Min Value	Max Value
IMAGE REF.	1037	0	mixed	R12-3347 p.100	R12-3352 p.99
ADDRESS	5638	2	mixed	1 Barlows Buildings	hargreaves Road
CENSUS PARISH	6	0	mixed	Cloughfold	Waterfoot
COUNTY	1	6276	letters	Lancashire	Lancashire
SURNAME	3425	0	mixed	ABBOTT	wotheriLL
FIRST NAME	3068	1	mixed	?	william
REL HSEHLD	48	0	mixed	ADOPTED DAUGHTER	WIFE
SEX	4	27	letters	f	w
AGE	116	0	mixed	1	not identified
OCCUPATION	3031	3268	mixed	(COMMON) BREWER	woollenWEAVER
BIRTH PARISH	2426	153	mixed	-	youngwood
BIRTH COUNTY	203	42	mixed	-	yorkshire

Table 3.7: Raw data quality analysis of 1891 census dataset.

Attribute	Unique	Missing	Type	Min Value	Max Value
IMAGE REF.	1083	0	mixed	r13_3846 p.10	r13_3850 p.99
ADDRESS	6363	1	mixed	1 & 3 double street	zechariah scarr
CENSUS PARISH	2	0	mixed	rawtenstall	rawtenstall"
COUNTY	1	0	letters	lancashire	lancashire
SURNAME	2725	0	mixed	?	zucharaloving
FIRST NAME	3658	1	mixed	?	zubar
REL HSEHLD	69	5	mixed	?	wifes sister
SEX	5	2	mixed	d	s
AGE	136	0	mixed	1	un 1m
OCCUPATION	4574	0	mixed	-	yeast traveller
BIRTH PARISH	2471	0	mixed	-	york
BIRTH COUNTY	100	48	mixed	-	yorkshire

Table 3.8: Raw data quality analysis of 1901 census dataset.

Take Table 3.3 as an example, which shows the analysis results of the 1851 census returns. There are many missing, abbreviated, inconsistent, and wrong values in the table. For example, the “ADDRESS" attribute has 318 missing values. The “SEX" attribute has three values, this contradicts the common sense of male and female genders. The analysis of the results show that the third value in the 1851 census data is “j", which is a wrong value. The “AGE" attribute has 129 unique values, some are normal digits, and some come from the nonstandard expression, such as “1M". All these mistakes extracted in the data analysis step will affect the data linkage process and should be corrected before the actual data linkage step is started.

We also notice that for the 17,033 records collected from 1851 census, there are only 1,009 different surnames and 999 different first names, which means many common names appear in the data. An analysis shows that many common names, such as “ASHWORTH" for surname and “JOHN" for first name occurred in the region. Figure 1.1 shows the percentage of the top 1 and top 5 first names in the six historical census datasets. It suggests that names became more and more diversified over time, which also means that linking of the early censuses data is a more challenging task. The type of each attributes in Table 3.3 indicates that many attributes contain mixed values. For example, the “AGE" attribute contains entries in both digital form (e.g., numbers) and other form (e.g. “un 1m"). This means multiple strings or unstandardized values exist.

In general, the quality of attributes in different census datasets varies across time. For example, the “ADDRESS" attribute has 318 missing values in the 1861 census data, but has zero or very few missing values in other datasets. The “OCCUPATION" attribute has thousands of missing values in the 1881 and 1891 datasets, but has few missing values in the other datasets. The quality of name attributes, however, are very consistent, with almost no missing values in each dataset.

3.3 Summary

This chapter introduces six historical census datasets used in our research and several key data preprocessing steps. We first describe the data collection background, which determines the data composition/structure as well as the quality problem arisen in the data collection steps. This is followed by the data analysis step which aims at identifying missing, wrong, inconsistent, and non-standardised attribute values. We summarises the statistics of these quality problems for each attribute across the six datasets. This analysis forms the basis for the data processing steps to be introduced in the next chapter.

Chapter 4

Historical Census Data Processing

In this chapter, we first describe data cleaning and standardisation techniques that have been applied to the historical census data. This step helps to significantly improve the data quality. Then we introduce two important data processing steps, household identification and record pair similarity calculation. These two steps generate basic information that will be used in the three historical census data linking methods that will be introduced in later chapters.

4.1 Data Cleaning and Standardisation

We employed data cleaning and standardisation techniques before the data linkage step for improving the quality of the data and formatting the data into a unified format. The data analysis step in Chapter 3 has identified missing, wrong, inconsistent values, and non-standardised values. Data cleaning and standardisation focus on how to eliminate these wrong and inconsistent values. This will greatly increase the likelihood of finding true matches. In our work, we implemented a 5-step approach as summarised below.

- Step 1: **Standardise data format**

  As shown in the data analysis step, some attribute values do not follow a standard format. For example, some surnames are in uppercase letters, while others are in lowercase letters. This may introduce errors in the linking step when string similarities have to be calculated. Thus, we standardised the values into

the same format: field names were standardised to uppercase letters, while attribute values were converted to lowercase letters.

- Step 2: **Remove non-meaningful values**

  Many non-meaningful values are in the data. These include symbols such as "=","?", and non-standard words, such as "no entry" and "not identified", to mark that the values that were not filled in the original census return. All these values are not useful and shall be removed before the linkage step. This was implemented by scanning through the datasets with domain knowledge and extracting useless entries and putting them into a lookup table as shown in Figure 4.1.

```
# Remove characters and words from input

[' ', '  ', '&','.','..', '...','[',']','<', '?','??',
'~','"',';', '^', '=','-', ',', 'empty house', 'uk',
'nk','(nk)', 'n k', 'n.k', 'n.k.', 'no entry','no
number','missing','*','\'','\\','/', 'not identified',
'not identifed', 'not specified','null','unknown', 'no',
'none','relationship not identified','relationship not
specified','not specifiednot specified','err:508','n...']
```

Figure 4.1: Values to be removed are a list of strings separated by commas.

- Step 3: **Split first name and middle name into two attributes**

  In the census data, the "FIRST NAME" attribute shows not only the first name of each person, but sometimes also contains middle names, or an abbreviated middle name after the first name. After analysis, we found examples in which the same person has a middle name in one census and no middle name in another census. This reduces the reliability of linkage between censuses. For solving this problem, we extracted the first string from the "FIRST NAME" attribute as the value of the "FIRST NAME", and put the other strings into a newly created "MIDDLE NAME" attribute. If there is no middle name in the "FIRST NAME" attribute, i.e., the first name is a sole string, the "MIDDLE NAME" attribute is blank.

Unexpected Age Values	Corrected Age
under 1m	0
< 1	0
< 1 m	0
< 2 m	0
un 1m	0
12m	0
1.25	1
1.5	1
1.75	1
< 2	1
3.5	3
3.7	3
3.75	3

Table 4.1: Age standardisation

- Step 4: **Age standardisation**

  Analysis results show that the “AGE" attribute contains mixed type of data. Although the age values for most adults are in numbers, the age format for infant varies, for example, “11 d", “12 w", and “UNDER 1M", which means “11 days", “12 weeks", and “under 1 month". In order to unify the age format into a number format only that represents age as a number of years, we developed an approach to automatically find and convert these non-digital values to standard digital values. All values ending with “d" or “w" were automatically converted to “0". Values ending with “m" or more complex non-digital values were put into a lookup table as shown in Table 4.1, which facilitates the format conversion.

- Step 5: **Fill entries with correct values**

  This step was done by exploring the relationship between attribute values using a rule-based method. An example is the cleaning of gender values. Each entry was validated using the relationship to the household head and its first name. When such a household relationship was used, a matching table in Figure 4.2 was defined to map the relationship to gender options, such as “daughter" to

```
# relationship vs gender

"m" := ["nephew", "niece's son", "father", "boarder's son", "grand father", "brother",
"lodger's grand son", "husband", "adopted brother", "servant's grand son", "step
brother", "great nephew", "brother's son", "relation's son", "servant's son", "shopman",
"step son", "son", "grand son in law", "visitor's son", "step father", "uncle", "adopted
son", "father in law", "lodger's son", "son in law", "great grand son", "grand nephew",
"brother in law", "grand son","ostler"]

"f" := ["step sister", "maid", "great niece", "lodger's wife", "step daughter", "visitor's
daughter", "lodger's daughter", "foster daughter", "niece's daughter", "brother's wife",
"step mother", "boarder's daughter", "niece in law", "wifes sister", "adopted daughter",
"servant's daughter", "sister", "daughter", "brother's daughter", "grand daughter",
"step aunt", "great aunt", "aunt in law", "inmate daughter", "visitor's wife", "servant's
grand daughter", "grand niece", "lodger's niece", "sister in law", "niece", "grand
mother in law", "companion", "daughter in law", "great grand daughter", "half sister",
"grand mother", "wife", "mother in law", "aunt", "mother", "boarder's wife",
"governess","house keeper","spinster"]
```

Figure 4.2: Gender and corresponding relationships.

"female". A problem here is that some names are unisex. In these cases, this dependency was not used as the decision rule, and the attribute values remain unchanged.

Some cleaning tasks require sophisticated domain knowledge. For example, the classification of occupations follows a strict taxonomy [120]. Understanding such taxonomy requires expertise in historical census data and occupations. Furthermore, different "OCCUPATION" values may correspond to the same occupation. This makes the cleaning task further complicated. Although preliminary automatic occupation approach has been reported by Naive Bayes classifier with feature selection [121], it is not accurate enough and will introduce more errors into the data. To prevent introducing errors into the data cleaning step, we kept the occupation values unchanged. We leave it a future work to develop more advanced automatic methods for coding historical occupation descriptions for record linkage.

After cleaning, the unique values in the “SEX" attribute are reduced from three to two. The range of “AGE” values is changed to 0 to 92 years in digits. The type of “FIRST_NAME" is changed to “letters". This shows that the cleaning and standardisation can significantly improve the quality of the data. Basic statistics on cleaned and standardised data of the 1851 census dataset are shown in Table 4.2. Note that two new attributes have been added into this table. “MIDD_NAME" stores the second string of a person’s first name from the “FIRST_NAME" attribute. “HSEHLD_ID" stores a unique ID for each household, whose details will be described in Section 4.2.

Raw data analysis result					
Attribute	Unique	Missing	Type	Min Value	Max Value
IMAGE REF.	915	0	mixed	ho2248_1 p.10	ho2249_2 p.99
ADDRESS	509	318	mixed	-	yewin hill
CENSUS PARISH	10	0	mixed	Cloughfold	Waterfoot
COUNTY	1	0	letters	Lancashire	Lancashire
SURNAME	1009	0	mixed	-	YOUNG
FIRST NAME	999	0	mixed	-	ZILPOH
REL HSEHLD	60	0	mixed	ADOPTED DAUGHTER	WIFE'S NIECE
SEX	3	0	letters	f	m
AGE	129	0	mixed	1	1M
OCCUPATION	2739	22	mixed	-	YENTERERS WIFE
BIRTH PARISH	1407	5	mixed	-	yovgend
BIRTH COUNTY	65	3	mixed	-	yorkshire
Cleaned data analysis result					
Attribute	Unique	Missing	Type	Min Value	Max Value
IMAGE REF.	915	0	mixed	ho2248_1 p.10	ho2249_2 p.99
ADDRESS	414	874	mixed	1 back mill gate	yewin hill
CENSUS_PARISH	10	0	mixed	cloughfold	waterfoot
COUNTY	1	0	letters	lancashire	lancashire
SURNAME	995	5	mixed	ains	young
FIRST_NAME	632	11	letters	a	zilpoh
RELATION_TO_H_OF_HSEHLD	53	23	mixed	adopted daughter	wife's niece
SEX	2	0	letters	f	m
AGE	90	1	digits	0	92
OCCUPATION	2734	4708	mixed	12 scholar and 12 piecer	yenterers wife
BIRTH_PARISH	1384	311	mixed	accrington	yorgend
BIRTH_COUNTY	49	7	mixed	at sea	yorkshire
MIDD_NAME	23	15764	letters	a	z
HSEHLD_ID	3295	0	digits	1	3295

Table 4.2: Data quality analysis on the raw and cleaned data of 1851 census dataset. The number of missing values of some attributes increases after cleaning because many non-meaningful values in the original data have been removed.

4.2 Automatic Household Identification

Census data were collected based on households. Generating household identifiers is the process of assigning a unique number to each household. It is an essential step towards data linkage because household identifiers provide a wealth of information for structural analysis of the household system and the changes in these systems over the period of time of our study.

We used a set of rules to perform household identification. These rules are based on assumptions generated from domain knowledge. In a census spreadsheet, the value for the “RELATION_TO_H_OF_HSEHLD" attribute for each household should start by the head of the household. Based on knowledge obtained from social scientists, there are four possible values for the head of the household in the UK census data, namely “head", “head of family", “widow", and “widower". We have developed an algorithm to scan through a census data file. Each time a record has one of these head of household role values, the household ID (HID) is incremented by one, and this HID is assigned to all following records until another record with a head of household role is found.

We compared the automatically detected HIDs against the manually labelled results provided by a domain expert. Table 4.3 shows the differences between results on all six historical census datasets. It can be observed that the proposed method is very effective when combined with domain knowledge, with more than 99% HIDs correctly detected. This suggests that the proposed HID detection method can be used to replace the manual labelling, which greatly reduces the manual data cleaning efforts by social scientists. We also investigated the reason for the difference between the results from our method and from domain expert. It turns out that this is due to the errors in the relationship entries in the dataset. For example, there is a household with two “head"s because a married child is also considered as a head of the house-

Year	1851	1861	1871	1881	1891	1901
Number of Household	3295	4570	5576	6025	6378	6842
Number of Differences	1	2	10	0	0	26
Accuracy	99.97%	99.96%	99.82%	100%	100%	99.62%

Table 4.3: Accuracy of automatic HID detection on historical census datasets compared against manually labelled results.

hold. When the expert deals with such cases, they can check the original census image for confirmation, which is a function not included in our method.

The HID detection results can be used to further clean the census data. Here we aim at reducing the discrepancy of the addresses of members in a household. To do so, we checked whether the home addresses and surnames are consistent for the individuals in the same household (with the same HID). The results can be divided into the following four categories:

- same surname with same address,
- same surname with different addresses,
- different surnames with same address, and
- different surnames with different addresses.

The statistics on the distribution of these four categories are shown in Figure 4.3. A household with the same surname and same address suggests correct cleaning. The existence of households with different surnames may be because more than one family lived in a household, or the wife had remarried and brought her children from her previous marriage into the household, or visitors or helpers lived with a family. These do not necessarily suggest errors in the source data. Meanwhile, we are mostly interested in households with the same surname and different addresses. Our analysis shows that such case may be caused by two reasons. Firstly, it may due to different presentation of the same address. For example, in a household, one record has “carr" as its address value, but another record uses “carr head" as the value.

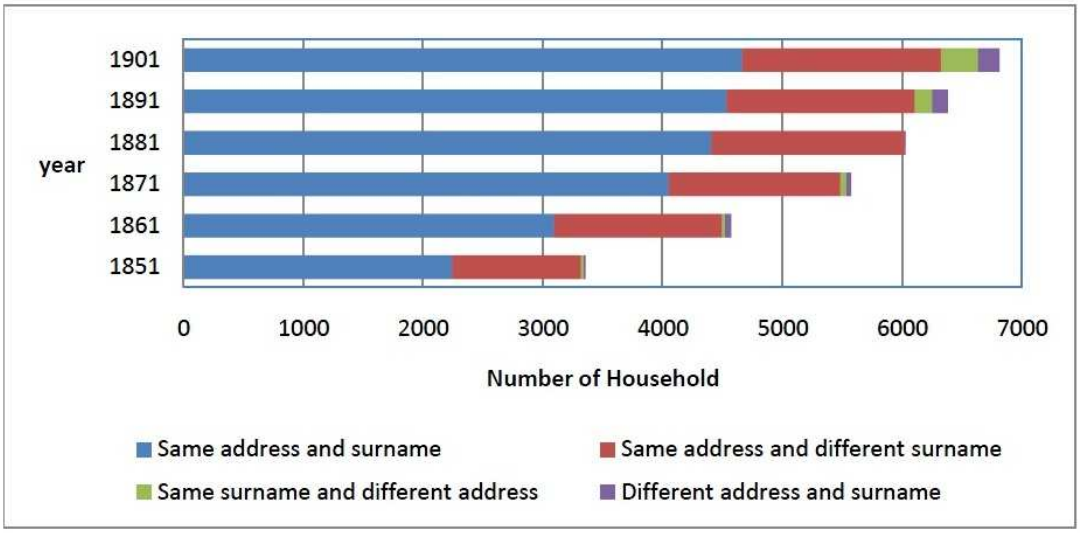


Figure 4.3: Distribution of household name and address relationships.

The same applies to “second row" and “second row front" in another household. In this case, the short address is converted into the long one as the long one may contain more complete information of the address. Secondly, the inconsistency on the addresses in the same household may come from errors in data entry during which operators had entered the data in a wrong row when the data were transformed from the original table. This makes the address of first or last record different from those of other records in the same household. In this case, the address that is common to the majority household members are used. However, this operation has to done prudently because we don’t want to introduce more errors into the data. For those households in which there are more than three different addresses or two addresses are split evenly among the household members, we left them unchanged because it is difficult to figure out which address shall be the correct one. For those households with two addresses, if the majority members have the same address, we change the address of other members to match the majority. This finalises the data preprocessing step.

4.3 Record Pair Similarity

The goal of record similarity calculation is to generate quantitative measurement of similarities between record attributes. We first applied blocking/indexing technique to subdivide the datasets into several blocks, so only records in the same block are compared. The blocking method can greatly reduce the number of record pairs to be compared and therefore speeds up the linkage process [61]. Secondly, we introduce how to select attributes and similarity functions, i.e., approximate string comparison methods, for attribute similarities calculation.

4.3.1 Blocking

Blocking was applied before the linking step because comparing records in two complete datasets can be time consuming. In our historical census datasets, the average number of records per dataset is around 20,000. If each record in one dataset is compared to all records in another dataset, about 400,000,000 record comparisons have to be performed. By simple analysis, it is easy to figure out that most comparisons are done between pairs of record with low similarity, which suggests that many comparisons are not necessary. To quickly exclude these record pairs while maintaining linking accuracy, we applied blocking/indexing technique before the linking step.

Blocking techniques divide a dataset into many small blocks so that each block only contains a limited number of records. These blocks are generated using criteria commonly known as blocking keys. The blocking keys split a dataset into blocks so that the comparison can be executed only between the records that have an identical blocking key. Noisy and low quality data may influence the blocking key generation. To tackle this problem, a blocking method shall reduce the possibility that records are inserted into wrong blocks [122]. Therefore, the first rationale to consider is that attributes with good quality shall be selected. These attributes should have few

wrong or missing values, and the values shall be more consistent with each other. Examples include “SEX” and names. The choice of encoding algorithms is also very important. Due to the phonetic nature of attribute values, we adopted phonetic algorithms such as “Double Metaphone", for encoding each selected attribute. They can well characterise variations and inconsistencies in spelling and pronunciation. For example, when using “Double Metaphone" encoding algorithm, “Smith" is encoded as “SM0" (primary code) and “XMT" (secondary code), “Schmidt" is encoded as “XMT" (primary code) and “SMT" (secondary code). Because both words have “XMT" in common, they are grouped into one block. To further improve the quality of blocking, we used multiple blocking keys to make sure that matched records have at least one blocking key in common.

In our research, we selected four key attributes to form blocking keys. They are “SURNAME", “FIRST_NAME", “ADDRESS", and “SEX". In order to avoid generating large size blocks, we combined two attributes in each blocking key generation. For example, the combination of “SEX" attribute which has only two values but with good quality, and “SURNAME" attribute generates good size blocks. We also generated multiple blocking keys to avoid missing any correct record pairs. The details of our blocking keys are:

- “Double Metaphone" encoded first three letters of the “SURNAME" attribute concatenated with the “SEX" attribute.
- “Double Metaphone" encoded first three letters of the “FIRST_NAME" attribute concatenated with “Double Metaphone" encoded first four letters of the “ADDRESS" attribute.
- “Double Metaphone" encoded first three letters of the “FIRST_NAME" attribute concatenated with “Double Metaphone" encoded first four letters of the “SURNAME" attribute.

To give a quantitative analysis on how blocking can improve the linking efficiency, we applied these blocking keys to all six historical census data linkage. Table 4.4 shows how many blocks were generated when using the aforementioned blocking keys, as well as the largest and average blocks in these generated blocks. Table 4.5 shows the total number of comparison reduction after using these blocking keys. Here we give an example on linking 1851 and 1861 historical census dataseta. These two datasets contain 17,033 and 22,429 records, respectively. The records are split into 11,991 small blocks, with the largest block containing 528 records and the average block length being 112 records. The total number of comparison is reduced from 382,033,157 to 2,441,819 which is equivalent to 99.36% reduction of the number of record pairs to be compared.

	1851-1861	1861-1871	1871-1881	1881-1891	1891-1901
Number of blocks	11,991	16,312	19,898	23,944	26,973
Largest blocks	528	513	600	618	571
Average blocks	112	100	104	87	73

Table 4.4: Number of blocks generated for pair-wise record linking of six historical census datasets.

	1851-1861	1861-1871	1871-1881	1881-1891	1891-1901
Number of comparisons before blocking	382,033,157	588,290,241	761,978,679	874,057,437	934,472,133
Number of comparisons after blocking	2,441,819	3,169,044	3,604,677	3,542,742	3,439,584
Reduction(%)	99.36	99.46	99.53	99.59	99.63

Table 4.5: Number of comparisons before and after blocking for pair-wise record linking of six historical census datasets.

4.3.2 Similarity Calculation

Computing record similarities is an important step in record linkage, which provides input to the record and household linking methods to be introduced in later chapters. Two factors to be considered in this step are choices of attributes to be compared and the similarity functions to be used for each selected attribute.

4.3.2.1 Attribute Selection

Not all 12 attributes in the historical census data are useful for record linking. For example, the “IMAGE REF" attribute shows the connection of the data in the spreadsheet with the original location of the scanned image. It does not contain any useful information on individuals or households. The values in the “COUNTY" attribute are all the same in six datasets, which means it is not useful either. The “CENSUS PARISH" attribute is a territorial unit. The numbers and names of parish changed significantly across time, so they cannot be used as a reliable attribute. The “OCCUPATION" attribute contains nonstandard values. There can be many values corresponding to the same occupation. Furthermore, “BIRTH PARISH", “BIRTH COUNTY" and the newly generated “MIDD_NAME" attributes contain many missing values, which will lead to problem in the similarity calculation. The values of “REL HSEHLD" changed significantly across time, so it is not selected.

We intend to use attributes with good data quality and that are highly informative. As a result, only the following five attributes, i.e., “SURNAME", “FIRST_NAME", “SEX", “AGE", and “ADDRESS", were selected as the key attributes for record pair similarity calculation. In these five attributes, “FIRST_NAME" and “SEX" are less likely to change across time though “FIRST_NAME" may be reported differently when, for example, a diminutive such as ’Liz’, ’Betty’, or ’Beth’ is used instead of ’Elizabeth’. “AGE" should normally accrue by 10 years (and never less than 9 or more than 11) between two consecutive censuses. “SURNAME" should only change

when a female marries, while “ADDRESS" may or may not change due to various reasons.

4.3.2.2 Approximate Similarity Measures

We calculated the similarity for these five selected attributes using Febrl [46]. For each attribute, one or more approximate string matching methods were applied, which leads to 10 combinations of attributes and approximate string matching methods that have been used to generate features of record pairs. A summary of these combinations is given in Table 4.6. Details of these approximate string matching methods and their implementation can be found in [123] and [46]. Each combination generates a similarity score between 0 and 1 for a particular attribute. The higher the score the more similar are the two attributes (scores of 1 indicate an exact match, 0 means no similarity).

Attribute	Method
Surname	Q-gram / Jaccard / String exact match
First name	Q-gram / Jaccard / String exact match
Sex	String exact match
Age	Gaussian probability
Address	Q-gram / Longest common subsequence

Table 4.6: Record similarity using five attributes and various approximate string matching methods [41].

By concatenating the similarity scores calculated for the six attributes shown in Table 4.6, a vector $Rs(r_{t,i,j}, r_{t',i',j'})$ can be got for record $r_{t,i,j}$ from one census dataset and $r_{t',i',j'}$ from another dataset. For convenience, we denote the similarity vector as $Rs(r, r')$. A total similarity score $Rsim(a, b)$ can be calculated by summing over the attribute-wise similarity scores. In Table 4.7, we show the distribution of $Rsim(a, b)$ on all six historical census datasets.

For $Rsim(a, b)$, the larger the similarity value, the more similar two records are.

Therefore, a simple way of finding matched record pairs is comparing the similarity $Rsim(a,b)$ against a predefined threshold ρ. If $Rsim(a,b) > \rho$, the record pair is considered to be a match, otherwise it is considered as a non-match. However, there are two problems with this simple method, which prohibits effective record linkage. Firstly, a number of factors may reduce the total similarity score between two records that belong to the same individual. Such factors include errors in the data, changes of addresses or surnames, and so on. Therefore, it is difficult to find an optimal ρ for this binary classification scenario. Secondly, the summed similarity score $s_{a,b}$ does not explicitly characterise the contribution of each attribute. In order to take the advantage of the discriminability of all attributes, we should use the full similarity vector, $Rs(r,r')$.

	1851-1861	1861-1871	1871-1881	1881-1891	1891-1901
$Rsim(a,b) \in [0,1)$	55	50	60	64	93
$Rsim(a,b) \in [1,2)$	7,915	13,746	19,722	23,762	24,639
$Rsim(a,b) \in [2,3)$	93,849	155,547	190,192	222,399	239,099
$Rsim(a,b) \in [3,4)$	183,833	292,214	321,173	302,352	347,352
$Rsim(a,b) \in [4,5)$	1,491,373	1,823,013	2,119,269	2,072,711	1,785,808
$Rsim(a,b) \in [5,6)$	359,131	493,595	585,703	605,559	643,909
$Rsim(a,b) \in [6,7)$	110,007	161,044	128,976	109,219	225,347
$Rsim(a,b) \in [7,8)$	182,867	212,858	220,815	187,648	149,484
$Rsim(a,b) \in [8,9)$	9,223	12,397	14,825	15,847	19,040
$Rsim(a,b) \in [9,10)$	2,979	3,572	2,985	2,453	3,490
$Rsim(a,b) = 10$	587	1,008	957	728	1,323

Table 4.7: Distribution of Similarity scores $Rsim(a,b)$ on six historical census datasets.

4.4 Summary

The purpose of Data cleaning and standardisation is to enhance data quality by fixing the problems identified in the data analysis step in Chapter 3. This is an important step towards accurate record and household matching between two datasets, which forms the basis for the rest of the chapters in this thesis. This chapter also introduces how to identify households in the data and assign them with unique household iden-

tifiers. Among other uses these HIDs are employed to clean the address attributes. Finally, we describe how to select attributes for record pair similarity calculation, and summarise the approximate string matching methods used for each attribute.

Chapter 5

A Group Linking Method for Household and Record Linkage

Household linkage and record linkage are two important tasks in our research. Successful household linkage is dependent on accurate record linkage which is often suffering from low data quality and ambiguous links of records. As a result, the accuracy of household links can not be guaranteed. In this chapter, we introduce an integrated solution based on group linking for both record and household linkage. The core idea is to treat a household as an integrated entity so as to explore group record analysis techniques to identify true household and record matches among many candidates. We consider the household as the basic reference group for linking every pair of individuals in two households across two census datasets, and then use the household linkage results to improve the pair-wise record linkage results.

The development of this approach is guided by domain knowledge, particularly, household information. This is due to the fact that census data were collected based on households. Therefore, using household identifiers is an essential step towards successful record linkage because household identifiers as introduced in Section 4.2 provide a wealth of information for structural analysis of the family system and the changes in these systems over the period of time of our study. In the linkage step, household information can significantly improve the linking accuracy. More detailed discussion about our approach is presented in the following sections.

5.1 Method Overview

The proposed linking method comprises six steps, as is illustrated in Figure 5.1. The input to the system are the two datasets to be linked, and the output are record and household pairs that have been classified as matches (two records or households belong to the same entity) or non-matches (two records or households do not belong to the same entity).

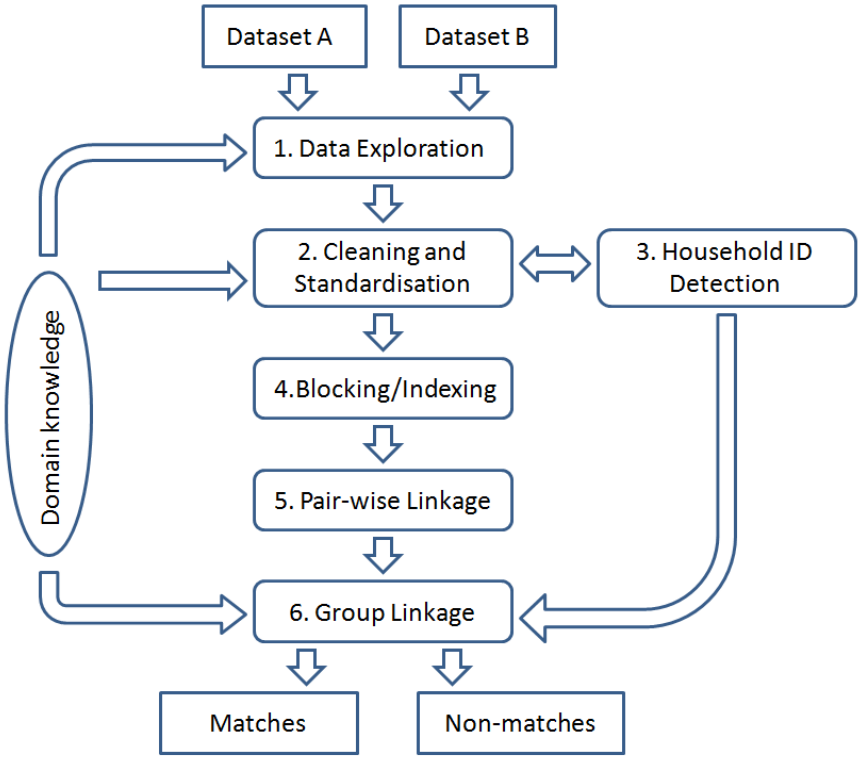


Figure 5.1: Flowchart of the proposed method [55].

The first step in the approach is data exploration. The purpose of this step is to obtain knowledge from the data so that they can be used in further data preprocessing. Details of this step have been introduced in Section 3.2.1. The second step is data cleaning and standardisation, which aims at reducing the errors and missing values in the data, as introduced in Sections 4.1. The third step is household ID detection, as described in 4.2, which assigns a unique Household ID (HID) to each

household, so as to facilitate the following record and group linking steps under the guidance of this domain knowledge. The fourth step is blocking/indexing, which helps to reduce the number of record comparison. This step has been explained in Section 4.3.1. The fifth step is to compute similarity scores for each pair of records under comparison. This step uses several measures to compute the similarities between individual attributes. The attribute similarities are concatenated into a vector which is then used in the following classification steps. Two record linkage classification methods are implemented using both a supervised approach, i.e., support vector machines (SVM), and an unsupervised approach, i.e., linked using a similarity threshold. These form the input to the final step, a group linking method, which is used to identify true household and record links among candidates. This allows a significant reduction of the number of ambiguous household and record links (one household or record from a dataset linked to more than one household or record from another dataset.

In the following sections, we will focus on the last two steps of the proposed method. Besides details of these steps, we will also address how to solve the problem of lacking ground truth for the unsupervised learning approach, and how to apply the SVM method using highly un-balanced training data. We will show that due to the characteristics of historical census data, domain knowledge can be used to improve both the efficiency and the accuracy of the linking performance.

5.2 Group Linking

Machine generated pair-wise linking results have been widely used by social scientists as the final outcomes of record linkage exercises, and they have used the results for further investigations [2, 3, 4, 5, 6, 7, 8]. The results normally contain large numbers of multiple record links that are ambiguous and that need further investigation before they can be accepted as a true match of a single record in one dataset

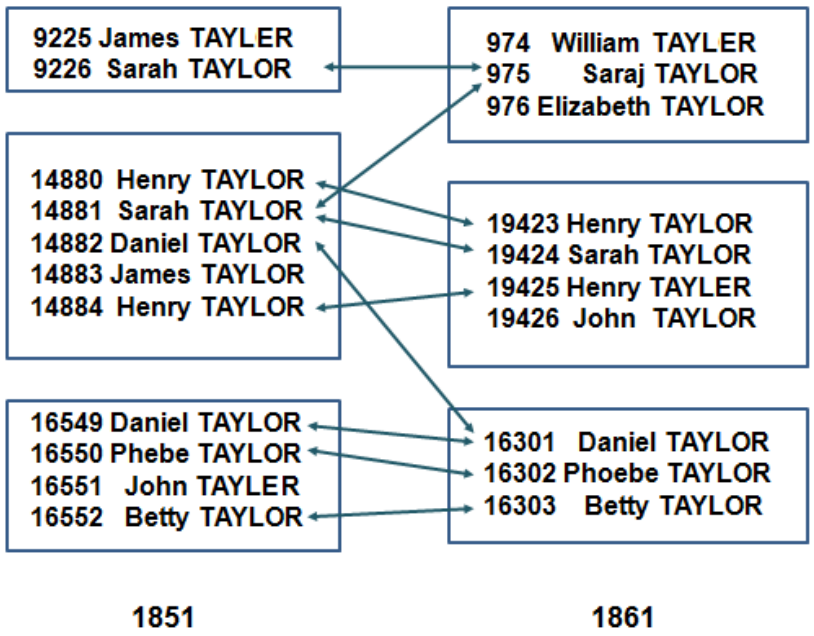


Figure 5.2: Illustrative example of multiple matches [10]. Each group of records corresponds to a household. The numbers are the record identifiers in each dataset.

to a record in another census dataset. This usually requires much manual effort to identify which linked record pairs are true matches, effort that is time consuming, cumbersome and error prone.

We argue that the problem of generating numerous multiple matches can be solved if the relationship between household members is taken into consideration. A simple example is shown in Figure 5.2 where “Sarah TAYLOR", in the middle household on the left-hand side, is matched to two records with a similar name in two different households on the right-hand side. In this example the middle household on the right-hand side is obviously the true match. Another example is “Daniel TAYLOR" in the lower right-hand panel, who is linked to two different individuals on the left-hand side with the true match to be found in the left-hand lower panel. In both cases the true match is obvious because they are identified in the context of other household members.

Based on this observation, we propose taking household information into account, so as to find the households which have a majority of their members matched. This allows several linked records belonging to the same household to be grouped, which is then used to calculate the best unique pairs in households to match across two census datasets. In this way household linking utilises richer information than would be used in standard pair-wise record linkage procedures, which leads to a refined selection of the correct record links while generating correct household links simultaneously. In this way we reduce linkage ambiguity and increase linkage accuracy.

5.2.1 Problem Definition

Let $H_{1,i}$ be the i^{th} household in the first census dataset $\mathcal{D}_1$, and $r_{1,i,j} \in H_{1,i}$ be the j^{th} record in this household, with $m_{1,i} = |H_{1,i}|$ be the number of records in household $H_{1,i}$, and $1 \leq i \leq m_{1,i}$. Similarly, let $H_{2,i'}$ be the i'^{th} household in the second census dataset $\mathcal{D}_2$, and $r_{2,i',j'} \in H_{2,i'}$ be the i'^{th} record in this household, with $m_{2,i'} = |H_{2,i'}|$ the number of records in household $H_{2,i'}$, and $1 \leq j' \leq m_{2,i'}$. The similarity vector $Rs(r_{1,i,j}, r_{2,i',j'})$ for a pair of records $r_{1,i,j}$ and $r_{2,i',j'}$ is calculated by concatenating all attribute similarity scores. The overall similarity score for record pair $r_{t,i,j}$ and $r_{t',i',j'}$ is $Rsim(r_{1,i,j}, r_{2,i',j'})$

Whether $r_{1,i,j}$ and $r_{2,i',j'}$ are a matched or a non-matched record pair can be classified by the pair-wise record linkage methods to be introduced in Section 5.3. Given $r_{1,i,j} \in H_{1,i}$, if only one $r_{2,i',j'} \in D_2$ is classified as a matched record to $r_{1,i,j}$, this record pair can be considered directly as a true match, and $H_{1,i}$ and $H_{2,i'}$ are matched households. On the contrary, if $r_{1,i,j}$ is matched to several records in $\mathcal{D}_2$, we have to determine which match is the true match. Therefore, the goal of the proposed group linking method is to determine which linked record in D_2 is the true match of $r_{1,i,j}$ when multiple matches are generated, and which household in D_2 shall be matched to $H_{1,i}$.

5.2.2 Ambiguous Link Reduction Method

To solve this problem, three strategies can be adopted. Firstly, we can remove multiple record links by simply choosing the matched pairs with the highest $Rsim(r_{1,i,j}, r_{2,i',j'})$ values for each $r_{1,i,j}$. This will generate either a unique record link, or multiple but less record links when several links have the same $Rsim(r_{1,i,j}, r_{2,i',j'})$ score for $r_{1,i,j}$. However, as we mentioned previously, due to erroneous data or changes in the data, exact matches are difficult to find, and $Rsim(r_{1,i,j}, r_{2,i',j'})$ may be low. Therefore, a true record match may not be at the top when ranked using $Rsim(r_{1,i,j}, r_{2,i',j'})$ only, and such a strategy will remove many true matches.

The second method is to set a similarity threshold ρ to help the decision. Record links with $Rsim(r_{1,i,j}, r_{2,i',j'}) < \rho$ can be removed from consideration. Even if such a threshold is set, one record in a dataset still can be linked to several records in another dataset, because the corresponding overall similarity scores $Rsim(r_{1,i,j}, r_{2,i',j'})$ are too close or identical.

Alternatively, as a third method, we can keep all record links in the group matching step. Because several linked records may belong to the same household, we calculate the best unique pairs of households that match across two census datasets, as detailed next.

Several group linking techniques have been proposed for bibliographic record linkage [11, 98, 124]. These techniques are based on unsupervised machine learning or graph-based approaches, which use information that explicitly connect records to collectively compute all links between records from two sets in an overall optimal fashion. Experimental studies have shown that these techniques can improve linkage quality significantly compared to traditional approaches that consider only pair-wise similarities between individual records. In our research, we extend the group linking

method proposed by On *et al.* to link two households [102]. For each pair of linked households, the household similarity score $Hsim(H_{1,i}, H_{2,i'})$ between two households $H_{1,i}$ and $H_{2,i'}$, can be calculated using the normalised weight of the matched individual record pairs in the two households:

$$Hsim(H_{1,i}, H_{2,i'}) = \frac{\sum_{(r_{1,i,j}, r_{1,i',j'}) \in \mathcal{M}} Rsim(r_{1,i,j}, r_{2,i',j'})}{M_{1,i} + M_{2,i'} - |\mathcal{M}|}. \tag{5.1}$$

where $\mathcal{M}$ is the set of record pairs classified as linked between $H_{1,i}$ and $H_{2,i'}$ according to the pair-wise linkage outcome, $M_{1,i}$ and $M_{2,i'}$ are the number of household members in $H_{1,i}$ and $H_{2,i'}$. This equation states that the household similarity is the sum of the similarities of matched record pairs normalised by the number of distinct members in these two households.

Here the record pair similarity function $Rsim(r_{1,i,j}, r_{2,i',j'})$ can take two forms. If a binary classifier is used to predict whether a record link is matched or non-matched, the similarity can take a binary form, such that

$$Rsim(r_{1,i,j}, r_{2,i',j'}) = 1 \tag{5.2}$$

if the record pair is predicted as matched by the pair-wise linkage method, or

$$Rsim(r_{1,i,j}, r_{2,i',j'}) = 0 \tag{5.3}$$

if the record pair is predicted as non-matched.

Alternatively, if the raw attribute-wise similarity is used, we have

$$Rsim(r_{1,i,j}, r_{2,i',j'}) = \mid Rs(r_{1,i,j}, r_{2,i',j'}) \mid_1 \tag{5.4}$$

where $\mid . \mid_1$ is the 1-norm of a vector, so that the overall record pair similarity is the

sum of the attribute-wise similarities. In the former case, the group linking reduces to computing the Jaccard index of two households [125]. The second form corresponds to solving a weighted bipartite matching problem [126].

Matched households can be classified by selecting the household links with the highest $Hsim(H_{1,i}, H_{2,i'})$ value. Here we assume that a household in one dataset can be matched to at most one household in another dataset. It should be mentioned here that this assumption does not always hold. The children in a household may get married or move out during the interval between two censuses. Therefore, a household can split into multiple households. However, as we mentioned at the beginning of this chapter, the purpose of household linkage is to find the households which have a majority of their members matched. Thus, our purpose is to link the most 'stable' part of households.

5.3 Pair-wise Record Linking

The group linking method proposed in Section 5.2 requires pair-wise record linking (shortened as "pair-wise" linking for convenience) results as input. Given attribute-wise similarity vector calculated from record pairs, we adopted two pair-wise linking methods to determine whether two records match or not. The first method addresses the problem when no training data are available. A similarity threshold method is adopted with the optimal threshold determined using the statistics from the linking results. The second method is a supervised learning method, for which a Support Vector Machine (SVM) classifier is learned to make predictions. All data are in the form of record pairs that are classified as a match or not [51].

5.3.0.1 Similarity Threshold-based Classifier

If we don't have the ground truth of the historical census data, in order to classify pair-wise linked records into a match or a non-match category, we apply a similarity

threshold method. As defined in Equation (5.4), by summing over the similarity scores in vector $Rs(r_{1,i,j}, r_{2,i',j'})$, we get a total similarity score $Rsim(r_{1,i,j}, r_{2,i',j'})$ which is a good measure to determine whether two records correspond to the same person. The larger the score, the more similar two records are. We compare the similarity score $Rsim(r_{1,i,j}, r_{2,i',j'})$ against a predefined threshold ρ. All pairs of records that meet the condition $Rsim(r_{1,i,j}, r_{2,i',j'}) > \rho$ are considered as matched pairs, otherwise they are considered non-matched pairs. This is the option that has been adopted by the group linkage method in [102]. Appropriate setting of the threshold ρ influences the accuracy of the final matching results. Further discussion on this topic will be given in Section 6.3.

5.3.0.2 Support Vector Machine Classifier

There are two problems with the similarity threshold method which prohibit effective record linking. Firstly, a number of factors may reduce the total similarity score between two records that belong to the same person. Such factors include, but are not limited to, errors in the data, changes of addresses or surnames, and so on. Therefore, it is difficult to find an optimal ρ for this binary classification scenario. Secondly, the overall similarity score $Rsim(r_{1,i,j}, r_{2,i',j'})$ does not explicitly characterise the contribution of each attribute.

To address these problems with the simple similarity threshold method, in this section, we use an SVM to classify the vectors $Rs(r_{1,i,j}, r_{2,i',j'})$ obtained from the record pair comparison step. Given the labelled binary dataset $(X, Y) = \{(\mathbf{x}_i, y_i)|i = 1, \ldots, N, y_i \in \{-1, 1\}\}$ (with class $y_i = 1$ being matches and class $y_i = -1$ being non-matches), where $\mathbf{x}_i$ are the indexed attribute-wise similarity vectors, i.e., $Rs(r_{1,i,j}, r_{2,i',j'})$, and y_i are their labels, an SVM classifier recovers an optimal separating hyper-plane $\mathbf{w}^T\mathbf{x} + b = 0$ which maximises the margin of the classifier. This can

be formulated as the following constrained optimisation problem [96]:

$$\min_{\mathbf{w},b,\xi} \quad \frac{||\mathbf{w}||^2}{2} + C\sum_i \xi_i \tag{5.5}$$
$$\text{s.t.} \qquad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \leq 1 - \xi_i \text{ and } \xi_i \leq 0$$

Here, a function ϕ is used to map the training vectors $\mathbf{x}_i$ into a higher dimensional space. $C = 2$ is the penalty parameter of the error term, and ξ is the margin slack variable. To handle the situation of imbalanced training data, we can assign a large penalty parameter for the matched class and a much smaller one for the non-matched class. All other parameters are tuned to optimal by cross correlation.

5.4 Implementation Details

We summarise our group linking approach in Algorithm 5.1. The input to the algorithm are all the matched record pairs $\mathcal{M}$ between the two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, and a household $H_{1,i} \in \mathcal{D}_1$. The output is the household $H^*_{2,i'} \in \mathcal{D}_2$ which has the highest similarity to $H_{1,i}$. From $\mathcal{M}$, we can find all records in $\mathcal{D}_2$ that match to records in household $H_{1,i}$. Each of these matched records belongs to a household in $\mathcal{D}_2$, and some of them might belong to the same household. To improve the efficiency of household matching, we then remove duplicate households, so that only unique household will be used to calculate the similarities to H^i_1 using Equation (5.1). Finally, the household(s) with the highest similarity $Hsim(H_{1,i}, H_{2,i'})$ will be selected as the output $H^*_{2,i'}$.

Step 4 in Algorithm 5.1 is important because it improves the efficiency of the proposed method. This is because several records in a household may be matched to other records that belong to the same household. Therefore, finding unique households will reduce the number of household similarity calculations. An example of this situation is shown in Figure 5.3. The four records in household A are matched

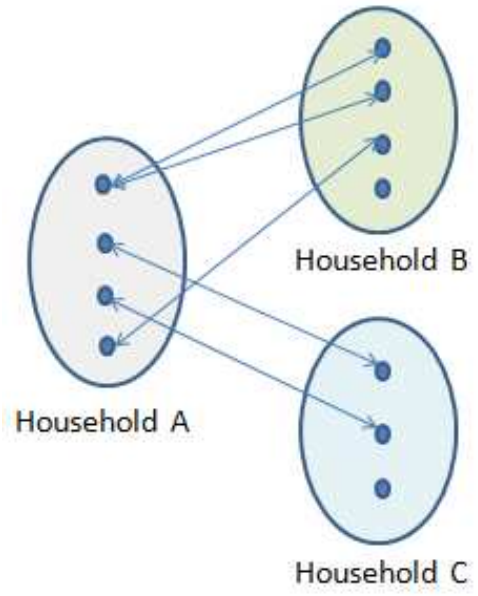


Figure 5.3: Example on household matching process.

to five records in households B and C. Instead of calculating household similarities five times, by finding the unique matched households, we only need to conduct two households similarity calculations. In this case, the number of household pairs to be linked is two.

5.4.1 Ground Truth Labelling

Because the datasets we have obtained do not contain the ground truth labels of which record pairs are matches or non-matches, we have manually identified true matching household and record pairs from the 1871 and 1881 census datasets. We chose these two datasets because they are the middle ones among the six datasets in our collection. Thus, we assume the sampled pairs have a similar distribution as household/record pairs sampled from the other pairs of datasets. The labelling process was based on randomly sampling households from the 1871 datasets. We then search through the 1881 dataset for matches. The comparison is based on matching the majority of household members, which is a strategy commonly taken by social scientist [44]. It also follows the rationale of the proposed group linking methods.

Algorithm 5.1: Household Linking

Input:
- Matched record pairs: $\mathcal{M}$
- All households in the second dataset: $\mathcal{D}_2$
- A household in the first dataset: $H_{1,i}$

Output:
- Best matching household: $H^*_{2,j} \in \mathcal{D}_2$

1: **for** $r_{1,i,j} \in H_{1,i}$ **do**
2: Find all matched records $\{r_{2,i',j'}\} \subset \mathcal{D}_2$ in $\mathcal{M}$
3: Find households $\{H_{2,i'}\} \subset \mathcal{D}_2$ for all $r_{2,i',j'}$
4: Find unique households $\{\tilde{H}_{2,i'}\} \subseteq \{H_{2,i'}\}$
5: Calculate household similarities $Hsim(H_{1,i}, H_{2,i'})$ for $H_{1,i}$ and $\{\tilde{H}_{2,i'}\}$ using Equation 5.1
6: Find $H^*_{2,j}$ with maximum $Hsim(H_{1,i}, H_{2,i'})$

Figure 5.4: Group linking method.

Once a matching household pair is identified, their corresponding matched records pairs are also identified. This allows us to generate both household and record level matched data. During this process, we have been helped by a social scientist, Dr. Mac Boots, who provided us with valuable domain knowledge. Nonetheless, the labelling process is a nontrivial task, due to the limited information that can be extracted from the data, high frequency of common names, and erroneous and nonstandard values in the data. This is especially the case for household pairs that have only one record matched. As a consequence, in our work, only record pairs that are clear matches are identified. In total, 1,250 matching household pairs and 4,808 matching record pairs have been identified.

Once the matched household and record pairs have been obtained, non-matched household pairs are generated by linking matched households to those that are not in the matched household pairs. This is done by randomly selecting household pairs containing at least one record link whose overall record pair similarity

$Rsim(r_{1,i,j}, r_{2,i',j'}) \leq 7$[1]. All record pairs extracted from these non-matched household pairs naturally form the non-matched record pairs. Due to the random selection, the total number of non-matches household pairs and record pairs vary.

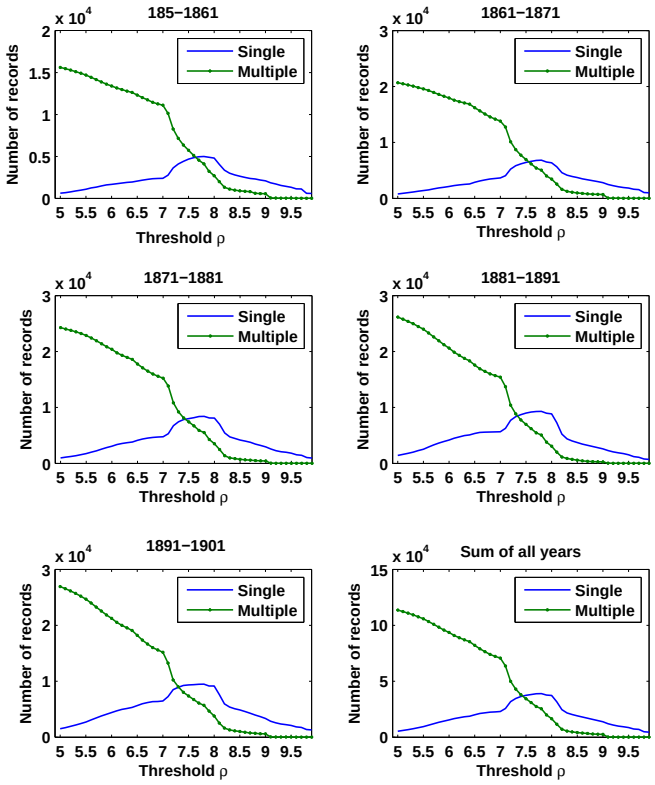


Figure 5.5: This figure shows the influence of different similarity thresholds (ρ) on the number of single matched and the number of multiple matched records. The single matched record thresholds are represented by solid curves and the multiple matched thresholds are indicated by lines with "*" on them. Here single matched records means one record in the first dataset is matched to only one record in the second dataset. Multiple matched records means one record in the first dataset is matched to more than one record in the second dataset. It can be seen from this figure that the data linking across different years follows the same trend.

[1]The highest record pair similarity is 10 due to the number of attribute/approximate string similarity measures used in our experimental studies.

5.5 Experimental Results

We have conducted experiments on all six historical census datasets following the steps introduced in the previous sections. We used LIBSVM [127] with an RBF kernel for training and testing of the record pair similarity vectors. To cope with the unbalanced data in the training set, we have set the penalty parameter for the matched class to 8 and for the non-matched class to 1, empirically. This is because the ratio between number of non-matched and matched record pairs are roughly $1:8$ in the training sets.

5.5.1 Results on Labeled data

In this section, we introduce experiments on both record linking and group linking on labelled data. The goal is to give quantitative evaluation of record linking methods based on the SVM and similarity threshold methods, and to compare several group linking strategies.

5.5.1.1 Record Linking

In the first experiment, we tested the performance of record linking methods. We start from an unsupervised setting based on the similarity threshold method. As described previously, setting a threshold separates linked record pairs into matched and non-matched pairs. A appropriate setting of the threshold influences the accuracy of the final matching results. To find the optimal setting we compare the linked results with respect to different threshold values ρ. Figure 5.5 shows the number of records in one dataset with “single matches" (i.e. a record in one dataset that is matched to only one record in another dataset) and with “multiple matches" (i.e. a record in one dataset that is matched to several records in another dataset), when different values for ρ have been set.

From Figure 5.5, two observations can be made. Firstly, for each pair of datasets,

there is a substantial portion of records in the first dataset that are matched to more than one record in the second dataset when ρ is small. With the increase of the value ρ, the number of multiple matches can be reduced but not eliminated. When the threshold ρ is set to a very high value, there are still a small number of multiple matches. For example, the linking result on the 1871 and 1881 datasets shows that when ρ is set to 9 out of 10, there are still 442 records that are linked with multiple other records. This result is in conflict with the domain knowledge that one record in one census dataset can only be matched with at most one record in another census dataset. This suggests that a large number of multiple matches are not correct matches, therefore, further processing is required.

The second observation is on the influence of the threshold ρ. Increasing ρ reduces the number of records with multiple matches. However, we found that many true links had been missed when ρ was set too high. For example, when ρ was set to 9, only 2,959 pairs of single matched records were found in the 1871 and 1881 pairwise linking result. However, our manually labeled data show there are at least 4,808 matched record pairs between the two datasets. This observation indicates that many matched record pairs are missing. On the other hand, when ρ is too low, a large number of multiple matches are generated. This makes it harder to separate the true match from these multiple matches.

To solve this dilemma, we analysed the number of records with exactly one match as a function of ρ. The curve in Figure 5.5 rises first, peaks in average at $\rho = 7.8$, and then drops. This is due to the fact that when ρ is small, many false matches are generated, and thus many records in one dataset are linked to more than one record in another dataset. With the increase of ρ, this number drops, so that the curve rises. When ρ becomes too large, many links are classified as non-matches, which causes the curve to drop again. When comparing curves of records with multiple matches,

the two curves in Figure 5.5 intercept at $\rho = 7.4$. This crossover point implies a balanced distribution for records with only one and with multiple matched records. This suggests that 7.4 could be a good candidate for the value of ρ for the group linking step. We use this intercept point as default threshold values for linking census datasets. Therefore, the threshold is set to 7.4 for all pairs of census datasets linkage because the interception points in all six pairs of datasets are remarkably consistent. We consider as matched only those record pairs whose total similarities are higher than this threshold. Table 5.5 (in page 88) "before group linking" shows the total number of matched record pairs between each pair of census datasets at the set threshold, as well as the number of distinctive records in the first dataset with single match and multiple matches. The process of setting ρ provides a new method to analyse linked results when ground truth is not available.

We also used the SVM method for pair-wise linkage in the case of a supervised learning setting. An SVM takes a set of input data as training set to train a model, and then applies the model to new record pairs to predict which of them are matches or non-matches. The input to the SVM model contains class labels and attributes vectors whose entries are the attribute-wise similarity scores calculated using the functions summarised in Table 4.6.

We performed an experiment to compare the effectiveness of two pair-wise linking methods. To build the training and testing sets for the SVM method, we have followed the sample generation method in Section 5.4. When generating the pair-wise link training samples, we have randomly selected pairs whose similarity is larger than 5. This is because linked record pairs with similarity lower than 5 are unlikely to be matched record links, so they are easy samples that do not contribute much discriminative information. We randomly split the labeled household pairs into a training set and a testing set with equal size. Based on the household-level training

Trainingset		Testingset	
Number of HHs	Number of Recs	Number of HHs	Number of Recs
22,039	40,317	22,370	40,659
22,863	41,719	21,536	39,261
21,941	39,575	22,457	41,405
22,322	40,567	22,081	40,413
22,290	40,072	22,116	40,908
22,205	38,843	22,179	42,137
22,968	41,889	21,447	39,091
21,528	38,919	22,872	42,061
22,539	41,590	21,865	39,390
22,293	40,938	22,111	40,042

Table 5.1: Numbers of household (HHs) and record pairs (Recs) in the 10 randomly split training and testing sets. They only contain record pairs whose similarities are equal to or higher than 5 out of 10.

and testing sets, we then built record-level training and testing sets. The number of household and record pairs in both training and testing sets, which are randomly generated ten times to conduct 10-fold cross validation, are summarised in Table 5.1.

For the SVM, the model parameters are tuned to be optimal on the training set via cross validation. For the similarity threshold method, we set the threshold $\rho = 7.4$ from the analysis of Figure 5.5. To evaluate the methods, we used accuracy, precision, recall, and F-score, which are defined as

$$accuracy = \frac{TP + TN}{TP + FN + TN + FN} \tag{5.6}$$

$$precision = \frac{TP}{TP + FP} \tag{5.7}$$

$$recall = \frac{TP}{TP + FN} \tag{5.8}$$

$$F = \frac{2 * precision * recall}{precision + recall} \tag{5.9}$$

where TP is the number of true positive, TN is the number of true negative, FP is the number of false positive, and FN is the number of the false negative. The F-score allows balanced contribution from both precision and recall.

The experimental results of two pair-wise linking methods are summarised in Table 5.2. Both the SVM and similarity threshold methods have achieved high classification accuracies, with 94.74± 0.73% and 84.14 ± 0.34% correct record links classified, respectively. Both recall scores are higher than 92%. This suggests that most true matches can be found. On the other hand, the precision of both the SVM and the similarity threshold methods is lower than 54%, which suggests that many non-matches have been classified as matches. This implies that the results contain large numbers of multiple matches that are ambiguous. Comparing these two methods, the SVM method has significantly outperformed the similarity threshold method in all evaluation criteria.

5.5.1.2 Group Linking

The pair-wise linking results show that there are many multiple matches, which lead to low precision scores. In order to reduce the number of multiple matches, we used the group linking method which refines the linking results based on household

	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
SVM method				
Before GL	94.74± 0.73	53.14 ± 3.53	94.57 ± 0.48	68.05 ± 0.84
After GL -Jaccard	99.11 ± 0.07	93.46 ± 0.95	91.29 ± 0.51	92.36 ± 0.67
After GL -Bipartite	98.68 ± 0.10	90.19 ± 1.18	87.05 ± 0.69	88.59 ± 0.87
Similarity threshold method				
Before GL	84.14 ± 0.34	26.12 ± 0.58	92.86 ± 0.42	40.77 ± 0.49
After GL -Jaccard	95.00 ± 0.18	56.59 ± 1.34	63.99 ± 1.18	60.06 ± 1.25
After GL -Bipartite	93.48 ± 0.18	45.03 ± 1.54	49.66 ± 1.55	47.23 ± 1.55

Table 5.2: Average pair-wise linking results on labelled data. Here, GL means group linking.

	SVM method	Similarity threshold method
Before group linking		
Total matched record pairs	4,101	8,334
Records with single match	1,970	1,384
Records with multiple matches	701	1,436
After group linking -Jaccard		
Total matched record pairs	2,314	2,663
Records with single match	2,252	2,015
Records with multiple matches	30	243
After group linking -Bipartite		
Total matched record pairs	2,286	2,587
Records with single match	2,216	1,841
Records with multiple matches	34	291

Table 5.3: Record level group linking results on labeled data.

similarity. For each candidate record with multiple matches, we compared their household similarities and only kept the record pair(s) with the highest household similarity. Following the description in Section 5.2.2, we applied Jaccard and Bipartite methods in group linking. The results on labeled data are summarised in Table 5.2. Compared with the results before group linking, both accuracy and precision scores are significantly improved. The increase on the precision scores implies that many duplicate matches have been removed. For the SVM method, the recall score dropped a little, but the same score for the similarity threshold dropped significantly.

The results in Table 5.2 shows that great improvements on the F-score have been achieved. This implies that group linking leads to an overall much better performance than the pair-wise linking method.

Details on the result analysis are shown in Tables 5.3 and 5.4, with the former showing the record linking results, and the latter showing the household linking results. Both tables give the number of single and multiple matches before and after the group linking step. It can be seen that the group linking method has greatly reduced the number of households and records with multiple matches. A large number of false positive classification results from the multiple matches have been removed.

	SVM method	Similarity threshold method
Before group linking		
Total matched household pairs	2365	5166
Households with single match	202	108
Records with multiple matches	423	517
After group linking - Jaccard		
Total matched household pairs	631	643
Households with single match	619	611
Households with multiple matches	6	14
After group linking -Bipartite		
Total matched household pairs	648	661
Households with single match	604	594
Households with multiple matches	21	31

Table 5.4: Household level group linking results on labeled data.

These results also show that the SVM is more reliable than the similarity threshold method. This is because the SVM takes advantage of the labelled attribute data to train the classifier but the similarity threshold method is an unsupervised classifier. Therefore, the SVM generates a better class separation plane. The results also show that Jaccard distance is a better measure than the Bipartite distance in group linking in delivering a larger number of matched record and household pairs and a smaller number of multiple matches.

5.5.2 Results on Historical Census Datasets

Finally, we show the record linking and household linking results across time in Tables 5.5 and 5.6. More intuitive results are plotted in Figures 5.6, 5.7, and 5.8 for the number of matched households, the percentage of reduction in the number of records with multiple matches, and the percentage of reduction in the number of households with multiple matches, when group linking is implemented with different distance measures and combined with the threshold and SVM record link classification methods.

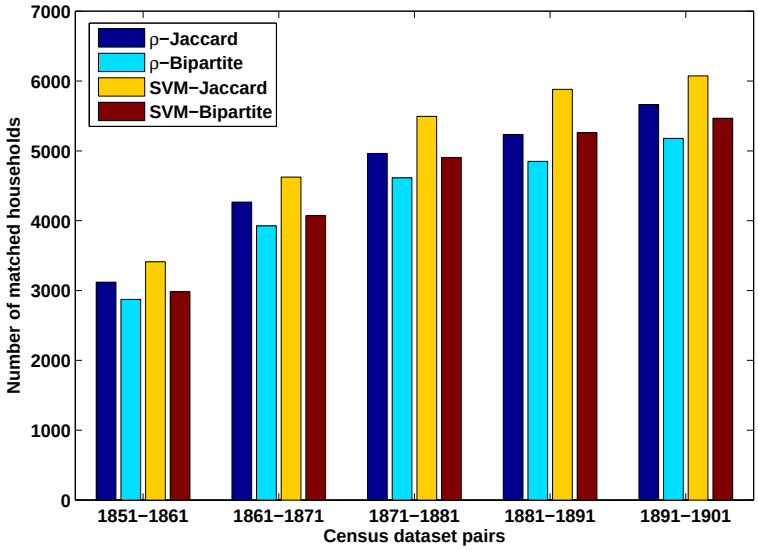


Figure 5.6: Number of matched households generated by different methods for the group linking step.

The results show that group linking can greatly reduce the number of multiple matches at both record and household link levels across all pairs of census datasets, with more than 85% reductions in both the numbers of records and households with multiple matches. Table 5.5, 5.6, and Figure 5.6 show that the numbers of matched record pairs and household pairs increase across time. This is natural because the number of households and records increase across time. On the other hand, Figures 5.7 and 5.8 show that the deduplication capability of the group linking methods is quite stable on all pairs of datasets linked, for each combination of distance measure and classification method.

Figures 5.7 and 5.8 also show that the difference between the Bipartite and Jaccard measures (defined in Section 5.2.2) is significant. Jaccard distance has generated a larger amount of total matched record and household pairs than Bipartite distance. This is consistent with the results on labeled data. On the other hand, Bipartite distance allows larger reduction in the percentage of record and household with multiple matches. This is because Bipartite distance uses the attribute-wise similari-

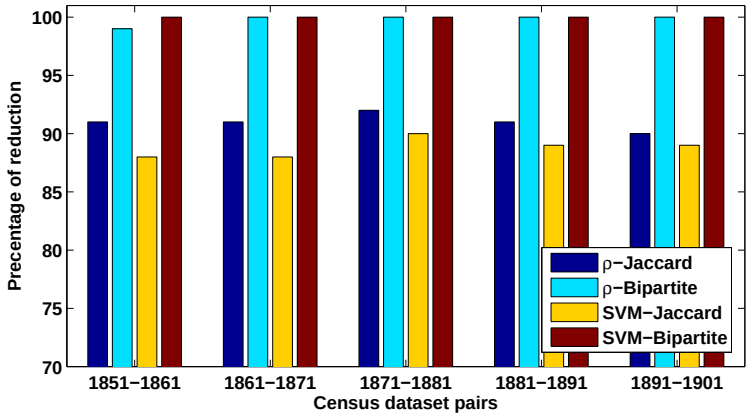


Figure 5.7: Group linking results shown as the percentage of reduction in the number of records with multiple matches.

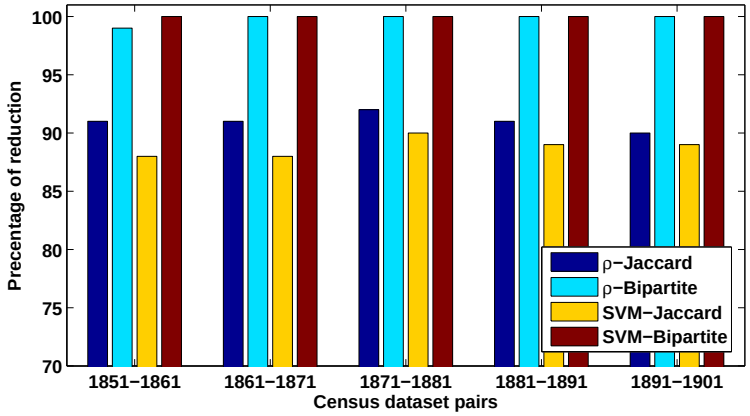


Figure 5.8: Group linking results shown as the percentage of reduced number of households with multiple matches in different methods.

ties which is more discriminative than the binary representation used by the Jaccard distance in the group linking step.

The SVM method consistently performs better than the similarity threshold method in both the number of matches, and reduction of multiple matches, which shows the superiority of the supervised learning method. It should be mentioned here that there are still records with multiple matches even after the group linking step. This is due to the fact that group matching of several households might have generated

the same similarity scores. In this case, it is hard to tell which household is the true match unless further analysis on the household and family is conducted. At this stage, we assume all household matches found are the true matches. We did not set a threshold to eliminate possible false matches as was done in [102]. The reason is that a household may change substantially between two censuses, for example, with children getting married and moving out, which greatly reduces the household match score. Therefore, a low household match score does not necessarily imply that two household are not matched.

Census dataset pairs	1851-1861	1861-1871	1871-1881	1881-1891	1891-1901
Before group linking (Similarity threshold method)					
Total matched record pairs	35,508	44,654	47,447	44,616	49,833
Records with single match	4,418	6,177	7,836	8,705	9,254
Records with multiple matches	6,358	7,733	8,131	7,774	8,025
After group linking - Jaccard (Similarity threshold method)					
Total matched record pairs	10,081	13,116	15,154	15,568	16,196
Records with single match	6,728	9,038	11,036	11,552	12,101
Records with multiple matches	1,049	1,285	1,276	1,216	1,388
After group linking - Bipartite (Similarity threshold method)					
Total matched record pairs	10,703	13,897	15,953	16,455	17,242
Records with single match	6,723	9,061	11,087	11,574	12,146
Records with multiple matches	881	1,065	1,037	978	1,092
Before group linking (SVM method)					
Total matched record pairs	48,943	60,400	70,177	69,631	63,101
Records with single match	4,163	5,692	7,249	8,237	9,041
Records with multiple matches	8,030	9,796	11,353	11,763	11,077
After group linking - Jaccard (SVM method)					
Total matched record pairs	11,073	14,225	17,346	18,577	18,723
Records with single match	7,421	10,099	12,698	13,629	13,953
Records with multiple matches	391	459	521	583	567
After group linking - Bipartite (SVM method)					
Total matched record pairs	12,165	15,482	18,602	19,989	20,108
Records with single match	7,420	10,067	12,685	13,580	13,893
Records with multiple matches	130	153	189	248	245

Table 5.5: Record linking results on six historical census datasets.

Census dataset pairs	1851-1861	1861-1871	1871-1881	1881-1891	1891-1901
Before group linking (Similarity threshold method)					
Total matched household pairs	24,865	31,068	32,826	30,674	34,502
Households with single match	448	647	966	1,184	1,251
Households with multiple matches	2,409	3,209	3,540	3,590	3,776
After group linking - Jaccard (Similarity threshold method)					
Total matched household pairs	3,121	4,266	4,963	5,236	5,663
Households with single match	2,634	3,643	4,318	4,516	4,782
Households with multiple matches	223	280	292	325	384
After group linking - Bipartite (Similarity threshold method)					
Total matched household pairs	2,874	3,927	4,615	4,850	5,180
Households with single match	2,841	3,919	4,605	4,832	5,152
Households with multiple matches	16	4	5	9	14
Before group linking (SVM method)					
Total matched household pairs	39,006	47,822	54,840	53,641	48,619
Households with single match	373	623	860	944	1,089
Households with multiple matches	2,601	3,448	4,046	4,314	4,375
After group linking - Jaccard (SVM method)					
Total matched household pairs	3,411	4,625	5,494	5,882	6,074
Households with single match	2,649	3,674	4,489	4,793	5,004
Households with multiple matches	325	397	417	465	460
After group linking - Bipartite (SVM method)					
Total matched household pairs	2,983	4,072	4,906	5,260	5,467
Households with single match	2,965	4,070	4,906	5,256	5,461
Households with multiple matches	9	1	0	2	3

Table 5.6: Household linking results on six historical census datasets.

	10 Years	20 Years	30 Years	40 Years	50 Years
ρ-Jaccard	166	163	178	118	2,232
ρ-Bipartite	169	180	206	142	2,160
SVM-Jaccard	165	152	163	203	2,291
SVM-Bipartite	159	171	199	258	2,187

Table 5.7: Households from the 1851 census that are linked over time periods with different lengths.

Finally, we show in Table 5.7 the number of households in the 1851 dataset that have only been linked in periods of different lengths. For example, the SVM-Jaccard method found 165 households only existed during the 1851-1861 period then disappeared in the 1871 census, but 2,291 households existed over the 50 years period from 1851 to 1901. The linking used the group linking results for each 10 year period reported above. For a household in the 1851 dataset, we first identified its match(es) in the 1861 dataset, then the match(es) in the 1871 dataset for each matched household in the 1861 dataset. The process continues iteratively until no match(es) can be found or until we have gone through all the datasets. All four methods have detected more than 2,200 households that have been linked over a period for 50 years. Only less than 200 households have disappeared every 10 years.

Such results may occur for two reasons. Firstly, the group linking is based on the record linking step. As long as record matches can be found for a member in a household for a 10 year period, the household linking continues for the next 10 year period. This means even if members in a household have perished or moved away, the linking process can be continued if at least one household member can be found in the following census datasets. The fact that a large number of household links has been found for the whole 50 year period tells that some children in a household tended to stay in the same area as their parents even when they've grown up and formed a new family. Therefore, such a process has generated the possibility of tracing family trees. We plan to further manually evaluate these results with domain experts. Secondly, such results may also be due to the false matches in the record

linking step. Although it is hard to judge the correctness of such matches due to lack of ground truth information, this study provides social scientists with a means to trace household changes across time. As far as we know, this is the first work of this kind in the field of historical census record linkage.

5.6 Summary

In this chapter, we have introduced a group record linkage method to reduce ambiguous matches generated from pair-wise linking step by considering household information. The key idea is that members in a household are considered as a group. After similarities between record pairs are computed, they can be classified as matches or non-matches. These similarities and record link classification results can then be used to generate household linking similarities. Multiple record links that are presented in households with low similarities are then removed.

We have tried different options of pair-wise linking methods (using either an SVM classifier or similarity threshold method) to automatically link two consecutive historical census datasets across time. In the record pair comparison, selected attributes and similarity functions (distance metrics) are combined aims at finding record similarities.

We have tested our method on six Rawtenstall datasets The results show that the proposed method effectively reduces the number of multiple record and household matches and provide social scientists with a useful tool to process and analyse historical census data.

Chapter 6

Graph-based Household Matching

In the previous chapter, we have introduced a group-based household and record linking method. This method takes domain knowledge of household information into account to eliminate ambiguous links from the pair-wise linking step. The approach has shown to greatly reduce the number of ambiguous links and achieve highly accurate linkage results.

Nonetheless, household linking methods treat a household as a set of collected entities that correspond to individuals. They do not take the structural information of households into consideration. While personal information, such as marital status, address and occupation, may change over time, surnames of females may change after marriage, and even ages may change due to different time of the year for census collection or input errors, some aspects of the relationships between household members normally remain unchanged. Such relationships include, but are not limit to, age difference, generation difference, and role-pairs of two individuals in a household. If the relationship between the household members can be incorporated into the linking model, the linking accuracy can hopefully be improved.

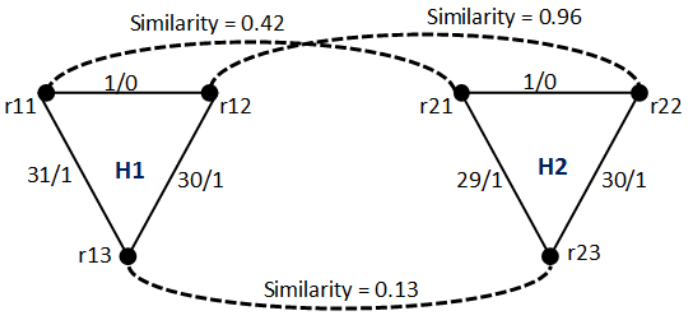
A graph-based approach is a natural solution to model the structural relationship between groups of records due to its capability to abstract complex relationships. In this chapter, a graph-based household matching method is introduced to explore the structural information in the household data. Our method treats members in a

household as the vertices in a graph, and uses edges to show the relational aspects of vertices. Then, the household linking problem is transformed into a graph matching problem. Household matching is determined not only by individuals, but also by the structure of the households they live in. We show that more intrinsic household information from the data can greatly improve the linkage accuracy.

The contribution of this chapter is two-fold. First, we develop a graph matching method to match households in historical census datasets. Our method demonstrates excellent performance in finding potential household matches and removing multiple matches. Second, to generate more accurate record matching results, we adopted a logistic regression method to estimate the probability that two vertices across two household graphs are matched. We evaluate our approach on both synthetic and real UK census data.

6.1 Method Overview

Given a candidate household from one dataset, the goal of our work is to find the best matching household among a list of target households in another dataset, and then to determine whether this match is correct. To achieve this goal, we propose a graph-based method to explore the household structure. We give an example in Figure 6.1 to illustrate how such structural information can be used for household matching. Figure 6.1 shows two households in two historical census datasets collected 10 years apart. Each household has three records. Records $r11$, $r12$, and $r13$ are in household $H1$, and records $r21$, $r22$, and $r23$ are in household $H2$. The details of several attributes of these records are shown in the figure, which include names, sex, age, and relationship to the head of household. The similarities of each record pair can be calculated using a logistic regression method whose details will be introduced in Section 6.1.2. From these similarities, it is very difficult to make a decision on whether these two households correspond to the same household or not. After



	ADDRESS	SN*	FN*	REL*	SEX	AGE
r11	goodshaw	trickettt	richard	head	m	32
r12	goodshaw	trickettt	elizabeth	wife	f	31
r13	goodshaw	trickettt	mary	daughter	f	1

H1

	ADDRESS	SN*	FN*	REL*	SEX	AGE
r21	goodshaw	trickett	richard	head	m	40
r22	goodshaw	trickett	elizabeth	wife	f	41
r23	goodshaw	trickett	m.	daughter	f	11

H2

*SN: Surname FN: First name REL: Relationship to head of household

Figure 6.1: An example of structural information of households extracted from two historical census datasets. The edge attributes are age differences and generation differences of connected vertices. Some attribute similarities between two pairs of records from two households are low. When the relationships between household members are considered, e.g. the roles in a household and age differences, it is clear that these two households shall be matched.

the structures of the two households are considered, e.g. roles of individuals in the households, the confidence that these two households are matched increases because the members show consistent roles and age differences in two households.

The steps of the graph-based household matching method is summarised in Figure 6.2. Given a candidate household and a target household to be linked, the first step is pair-wise linking, which generates attribute similarity for the records in the household pair. This step uses the approximate string matching methods that have been introduced in Section 4.3.2.2. Then a logistic regression method, whose model is learned from labelled training data, is used to generate the probability that a household pair is matched. This linking decision is made using a similarity threshold method, whose decision threshold can also be learned from the training data. Multi-

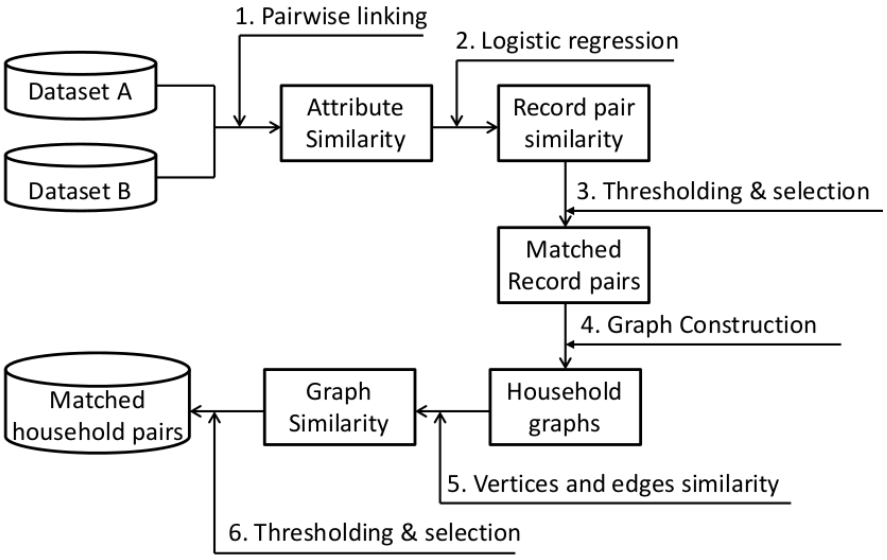


Figure 6.2: Key steps of the proposed graph matching method.

ple household matches can be generated from this step. Therefore, in the fourth step, a graph is constructed from each household that has been matched to the candidate household. The similarity between two graphs is calculated based on vertex and edge similarities, which are used to determine the true household matches. Details on the graph construction and matching steps will be explained in the following sections.

6.1.1 Definition

Here, we give formal definitions of the notion used in our method. Given a household H, an attributed graph [128] $G = (V, E, \alpha, \beta)$ can be defined, where $V = r_1, ..., i_i, ..., r_{|V|}$ is a set of vertices that correspond to $|V|$ household members. $E \in V \times V$ is a set of edges connecting every vertex pair, which show the relationship between household members. $\alpha = \{r_1, ..., r_i, ..., r_{|V|}\}$ and $\beta = \{e_{1,1}, ...e_{i,j}, ..., e_{(|V|-1),|V|}\}$ are the attributes associated with vertices and edges respectively. For convenience, and without confusion, we have used vertices and their attributes interchangeably. In a similar manner, we can define a graph $G' = (V', E', \alpha', \beta')$ on household H'.

Once these household graphs are built, the household linking problem becomes a graph matching problem, such that matched household can be identified based on graph similarity [129]. During this process, a key step is to generate a matching matrix for the graph pair such that vertices in G can be matched to vertices in G'. When labeled data are available, this problem can be solved by the quadratic or linear assignment method [130], which optimises an assignment matrix using the training data.

In the census household linking problem, domain knowledge tells us that each individual in one household can only be matched to one individual in another household. In the following, we show how this domain knowledge can be used to develop an efficient vertex matching method. Furthermore, we introduce a method to match household members prior to graph construction, such that the sizes of graphs can be reduced.

6.1.2 Record Similarity

This step takes attribute similarities between records, as generated in Section 4.3.2.2, as input. In the group linking method introduced in Chapter 5, each attribute makes the same contribution to the final linking step. However, in practice, each attribute may have a different contribution in determining whether two records are matched or not [20]. In order to estimate the contribution from each attribute, we model the vertex matching problem as a binary classification problem, and solve it using a logistic regression method. Assume we have T record pairs $\mathbf{x}_i$, $i = 1, 2, \cdots, T$, labelled as $y_i \in \{+1, -1\}$ for matched (+1) and non-matched (-1) classes. Let the features of record pairs be x_{ij}, where $0 \leq x_{ij} \leq 1$, $j = 1, 2, \ldots, Q$, and Q is the number of similarities generated from different approximate string matching methods on the

record attributes. A logistic regression model is given by

$$\log\left\{\frac{p(y_i \mid \mathbf{x}_i)}{1-p(y_i \mid \mathbf{x}_i)}\right\} = \mathbf{x}_i\mathbf{w} \tag{6.1}$$

where $\mathbf{w}$ is a vector of coefficients corresponding to the input variables. Then the maximum likelihood estimation of $\mathbf{w}$ is

$$\mathbf{w}^* = \arg\max_{\mathbf{w}}\left\{\sum_{i=1}^{T} -\log(1+\exp(-y_i\mathbf{x}_i\mathbf{w}))\right\} \tag{6.2}$$

which can be solved by iterative optimisation methods [131].

Once the optimal solution $\mathbf{w}^*$ is available, the posterior probability that a record pair is matched can be calculated as

$$P(y=1|\mathbf{x}_i) = \frac{1}{1+\exp(-\mathbf{x}_i\mathbf{w}^*)} \tag{6.3}$$

Note that this posterior probability is considered as the vertex similarity in the following graph model. Given $\mathbf{w}^*$ and the similarity vector $\mathbf{x}$ calculated between a pair of records, this equation leads to the similarity scores between nodes as illustrated in Figure 6.1. It should also be pointed out that the logistic regression based record similarity is independent of graph matching, hence, can be used on any pair-wise record comparison so long as a training set is available.

6.1.3 Record Linking

The outputs of the above step are record pair similarities. Here, we need to determine which record pair may be a correct match. Decisions can be made by comparing the vertex similarity with a threshold ρ, such that

$$P(y=1|\mathbf{x}_i) > \rho \tag{6.4}$$

In our method, following the classic decision rule of logistic regression classification [51], we set $\rho = 0.5$.

After comparing the record pair similarity with the threshold ρ, low similarity record pairs are removed from consideration. In the remaining record pairs, a candidate record may still be linked to multiple target records. For this case, the record pair with the highest similarity shall be selected as the match. In some cases, more than one record pairs may have the same highest similarity value, then all of the matched records are selected as the matching outcome.

6.1.4 Graph Generation and Vertex Matching

After the record pair selection step, a graph can be generated for each household. Note that the record linking step can remove a large number of links with low similarities, such that individual links in a household without high similarity do not need to be included in the graph generation. This allows small household graphs to be generated, which leads to high computation efficiency.

As mentioned previously, several target records may be matched to a candidate record in the record linking step. Therefore, one-to-many and many-to-one vertex mappings may be generated between two household graphs. Then the optimal vertex-to-vertex correspondence has to be determined. Although such vertex matching can be done in a supervised learning manner [130], in our method, we adopted the Hungarian algorithm [132], which is a more straightforward method with an $O(n^3)$ computational complexity, where n is the number of vertices. This algorithm generates the vertex matching that maximizes the sum of matched probabilities. The output of this step are graph pairs with one-to-one vertex mappings.

6.1.5 Graph Similarity and Matching

In the previous record linking step, a record may be linked to multiple records in different households. Therefore, a graph containing the record may be linked to several other graphs. Similar to the record linking step, decisions also have to be made about which graph pair is the most likely correct match, and if there are multiple matches, which pair is the correct one. This requires the calculation of graph similarity. Here, we define the similarity between graph G and G' as

$$f(G, G') = \lambda f_v(V, V') + (1 - \lambda) f_e(E, E') \tag{6.5}$$

where $f_v(V, V')$ and $f_e(E, E')$ are the total vertex similarity and total edge similarity, respectively, and λ is a parameter that control the contribution from $f_v(V, V')$ and $f_e(E, E')$.

Note that vertex similarity has been generated in the record linking step from the output of the Hungarian algorithm. Let $sim_v(r_i, r'_{i'})$ be the similarity of vertices r_i in graph G and a matching record $r'_{i'}$ in G', and the total number of matched vertices between G and G' be $\mathcal{M}$, then

$$f_v(V, V') = \frac{\sum_{i=1}^{\mathcal{M}} sim_v(r_i, r'_{i'})}{\mathcal{M}} \tag{6.6}$$

The calculation of total edge similarity is based on differences of edge attributes (details to be described in Section 6.2) between each pair of edges in the graph pair. Let $e_{i,j,k}$ be the $k^{th}(k \in [1, ..., K])$ attribute of the edge $e_{i,j}$ which connects record r_i and r_j in graph G, and $e'_{i',j',k'}$ be the corresponding edge in graph G', then

$$f_e(E, E') = \frac{\sum_{i=1}^{|E|} sim_e(e_{i,j}, e'_{i',j'})}{|E|} \tag{6.7}$$

where $|E|$ is the number of edges in the graph. $sim_e(e_{i,j}, e'_{i',j'})$ is the edge similarity,

which is defined as follows

$$sim_e(e_{i,j}, e'_{i',j'}) = \frac{\sum_{k=1}^{K} \tau_k sim_a(e_{i,j,k}, e'_{i',j',k'})}{K} \tag{6.8}$$

where $sim_a(e_{i,j,k}, e'_{i',j',k'})$ is the attribute similarity, and τ_k is the weight assigned to each attribute. In our method, we set equal weights for the attributes.
The graph similarity calculation allows selecting the optimal match of graph G from several target graph candidates. Then whether the selected graph $G^{'*}$ is a correct match of the candidate or not can be judged by the following condition:

$$f(G, G^{'*}) > \eta \tag{6.9}$$

If the graph similarity is larger than threshold η, then it is considered as a correct match. Note that parameters λ, τ and η can be learned from the training set by grid search. Such a search is implemented as an exhaustive searching through a subset of the hyperparameter space of these three parameters, which is guided by cross-validation on the training set [133].

6.2 Implementation Details

In this section, we give implementation details of several key steps in our method. Starting from the record similarity calculation, as introduced in Section 6.1.2, we adopted 10 combinations of attributes and approximate string matching methods to generate features of record pairs for the logistic regression model. In calculating the total vertex similarity $f_v(V, V')$, an alternative method is the group linking approach proposed in [102]. We implemented this model and combined it with the total edge similarity for graph similarity calculation. Different from Chapter 5, we used the probability generated by the logistic regression step to calculate record similarity, instead of using empirical record similarity calculation by adding the attribute-wise similarities. Then the group linking based graph vertex similarity is calculated using

the following equation

$$f_v(V, V') = \frac{\sum_{i=1}^{L} sim_v(r_i, r'_{i'})}{M + M' - \mathcal{N}} \tag{6.10}$$

where M and M' are the numbers of household members in H and H' respectively. $\mathcal{N}$ is the set of record pairs matched between H and H' as defined in Equation (6.6). Note that different from the vertex similarity calculated in Equation (6.6), group linking takes the number of distinct household members into consideration rather than merely the number of matched members.

Equation (6.8) requires calculation of several edge attribute similarities. In the proposed method, such edge attributes are generated to reflect the structural property of households. In more detail, three attributes have been considered. They are:

- age difference between two household members connected by an edge;
- generation difference between two household members connected by an edge;
- the role pair difference between two household members.

The calculation of age difference is straightforward. When comparing edges in two graphs, the edge similarity on this attribute is the probability generated by the Gaussian distribution of the difference of the age differences in two edges.

$$sim_{age}(e_{i,j,k}, e'_{i',j',k'}) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(e_{i,j,k} - e'_{i',j',k'} - \mu)^2}{2\sigma^2}) \tag{6.11}$$

where μ and σ are the mean and standard deviation of age differences calculated from the household members in the training set.

The generation difference is based the relative generation with respect to that of the household head. A lookup table is built for this purpose. For example, a record with role value "wife" is in the same generation as the a record with "head", therefore their generation difference is 0. The generation difference between "head" and "son" or

"daughter" is 1, and between "head" and "grandson" or "grand daughter" is 2. The role pairs are even more complex. We investigated the possible role pairs between two household members and generated a lookup table to show how such role pairs can change. For example, "wife-son" may change to "head-son" if the husband of the household has died between two censuses and the wife has become the head.

When comparing two edges, binary values are generated for both generation difference and role pair attributes. If the corresponding generation difference value of two edges is different, the similarity is 0, otherwise, it is set to 1. For the role pair attribute, if a role pair change is has been recorded in the training data, we set the similarity to 1, otherwise, it is set to 0.

6.3 Experimental Results

The proposed method is validated on both labeled data and complete historical census data. The complete data are the six census datasets collected from the district of Rawtenstall in North-East Lancashire in the United Kingdom, for the period from 1851 to 1901 in ten-year intervals, as introduced in Chapter 3.

6.3.1 Results on Labeled Data

Same as in Section 5.4.1, we have manually labeled 1,250 matched household pairs from the 1871 and 1881 historical census datasets. The labels also include matched records in the matched households. These became the positive samples in the dataset. Then we extracted negative samples by randomly selecting households and their records in the 1871 and 1881 datasets, which links to the labelled positive households. Because both household and individual records follow one-to-one match restrictions, we are sure that these negative samples are true non-matched samples. In this way, we have generated a dataset with ground truth at both household and record levels.

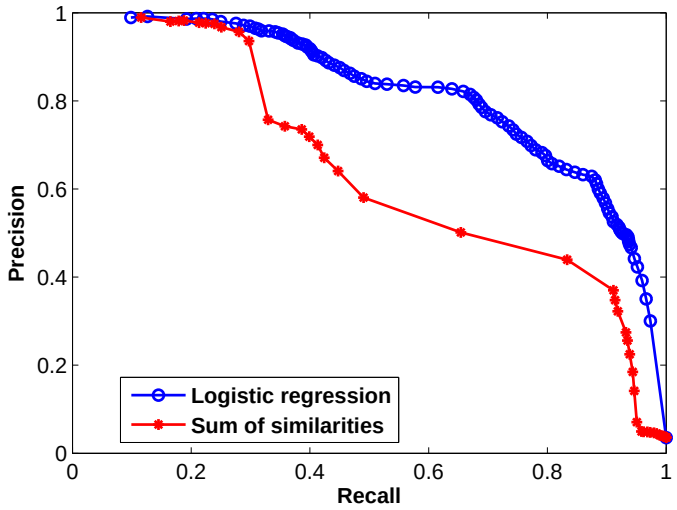


Figure 6.3: Precision-recall curve from logistic regression and sum of similarities for record linking.

We split the labeled data into a training and a testing set with equal number of households. We trained the logistic regression model in Equation (6.1) using the training set, and recover λ in Equation (6.5), τ in Equation (6.8), and η in Equation (6.9) by cross validation on the training set. These parameters were then applied to the graph matching model which was evaluated on the testing set. Each experiment was performed for 10 times, with randomly split training and testing sets.

The first step of a graph construction is to select records which are used as the vertices of graph. To do so, we used the proposed logistic regression method and the sum of the attribute-wise similarities generated by the approximate string matching methods [55] to calculate the similarities between record pairs. Then we applied a similarity threshold method to select those record pairs whose similarities are above the threshold as matched record pairs. To evaluate the effectiveness of the logistic regression and similarity threshold methods, we have generated the precision-recall curve as defined in Section 5.5.1.1 to show their performance with different threshold values. The results are displayed in Figure 6.3. This figure shows that the performance of the logistic regression model significantly outperforms the similarity

	Number of records	Number of households
Average testing set size	40,537	22,103
After logistic regression (LR)	8,473	5,205
After similarity threshold (ST)	2,620	1,297
Difference between LR and ST	5,853	3,908

Table 6.1: Average number of record pairs and household pairs in 10 testing sets before and after the logistic regression and similarity threshold methods are applied.

threshold method. This is due to the training process of logistic regression that allows better modelling of the data distribution.

Table 6.1 shows the average number of record pairs and household pairs in 10 testing sets before and after the logistic regression and similarity threshold methods are applied. Both approaches can greatly reduce the number of record and household pairs to be considered in the graph construction. The logistic regression has demonstrated superior performance than the similarity threshold method in generating by average 5,853 fewer record pairs and 3,908 fewer household pairs,respectively. This means not only the number of graph pairs to be compared, but also the size of each graph, will be much smaller, This leads to great improvement of the efficiency in the graph matching step, which is very important when we are dealing with large historical census datasets.

In order to evaluate the household matching performance, we compared the proposed method (Graph Matching) with several baseline methods. The first baseline method (Highest Similarity) matches household based on the highest record similarity. If one candidate household is matched to several target households, the target household with the highest record similarity is selected. The second baseline (Vertex Similarity) builds household graph using linked records. Then the household matching is determined only by the vertex similarity calculated by Equation (6.6). This is equivalent to calculating the mean record similarity on those records used to build graphs. The third method (Group Similarity) is the group linking method [55] as

	Accuracy (%)	Precision (%)	Recall (%)	F-score (%)
Similarity threshold method				
Highest Similarity	98.42 ± 0.06	70.88 ± 1.21	74.61 ± 1.43	72.69 ± 1.30
Vertex Similarity	98.25 ± 0.06	68.46 ± 1.23	70.51 ± 1.35	69.47 ± 1.28
Group Similarity	99.78 ± 0.02	96.07 ± 0.45	96.18 ± 0.42	96.12 ± 0.43
Graph Matching	99.75 ± 0.03	95.54 ± 0.35	95.58 ± 0.68	95.56 ± 0.48
Group Graph	**99.84 ± 0.02**	**97.12 ± 0.42**	**97.23 ± 0.38**	**97.18 ± 0.40**
Logistic Regression method				
Highest Similarity	98.95 ± 0.04	80.99 ± 0.82	82.21 ± 0.85	81.59 ± 0.83
Vertex Similarity	98.79 ± 0.04	78.49 ± 0.78	78.94 ± 0.81	78.72 ± 0.78
Group Similarity	99.69 ± 0.04	95.27 ± 0.64	93.78 ± 0.65	94.52 ± 0.63
Graph Matching	99.68 ± 0.04	94.88 ± 0.66	93.74 ± 0.71	94.31 ± 0.67
Group Graph	**99.70 ± 0.04**	**95.57 ± 0.65**	**93.82 ± 0.70**	**94.69 ± 0.66**

Table 6.2: Comparison of household linking results on labelled data.

defined by Equation (6.10). We also replaced the vertex similarity with the group linking score in the graph matching step, so that the final decision of household matching is determined by the sum of group linking and edge similarity. We mark this method as (Group Graph). In each method, we have tried two record similarity calculation options, that is, the similarity threshold and the logistic regression. The values of key parameters λ, τ, and ρ are tuned to optimal for each methods. We then evaluated the methods on the testing sets using accuracy, precision, recall, and F-score as the criteria.

The experimental results are summarised in Table 6.2. In general, all methods have achieved very high accuracy. This is due to the correct classification of large amount of non-matched household pairs which occupies a majority of the data to be classified. For precision and recall, the group similarity, graph matching, and group graph approaches have shown similar performance in all evaluation criteria. In particular, the graph matching method has generated the best accuracy, precision, recall, and F-score when combined with group similarity for graph similarity calculation. On the other hand, the methods merely using the highest or average vertex similarity performed much worse than the other three group or graph based methods. This

shows that by considering the structural information of households, we can greatly improve the linking performance.

When performing vertex matching, the similarity threshold performed much worse than the logistic regression methods. This is due to the large number of false positive matches that have been introduced into the household matching step. However, it is interesting to see that when the graph method is added to link households, which takes vertex similarity as part of its input, the similarity threshold method generates slightly better performance than the logistic regression methods. The reason is that the graph-based method can remove a large number of households with multiple matches, which significantly increase the precision of the matching. On the other hand, the similarity threshold method tends to generate a much larger number of records than the logistic regression method in the record linking step. This leads to large numbers of households kept for the group linking, which makes probability of preserving correct household matches higher than that of the logistic regression method. In turn, a higher recall value can be generated. In summary, this observation also demonstrated the effectiveness of group linking and graph matching.

Finally, we give a quantitative evaluation on the contribution of both vertices and edges in the graph matching method. In this experiment, we have used only one set of training and testing sets. We tune the λ in equation 6.5 from 0 to 1. When $\lambda = 1$, the graph matching is based on the vertex similarity only. When $\lambda = 0$, the graph matching is solely dependent on the edge similarity. It can be seen from figure 6.4 that the results of using edge similarity only can generate higher F-scores than using the vertex similarity. This is because the household structure is less likely to change. When both vertex and edge similarities are combined, the F-score increases gradually, and peaks at $\lambda = 0.7$. This implies that both edge and vertex contain discriminative information for household linkage.

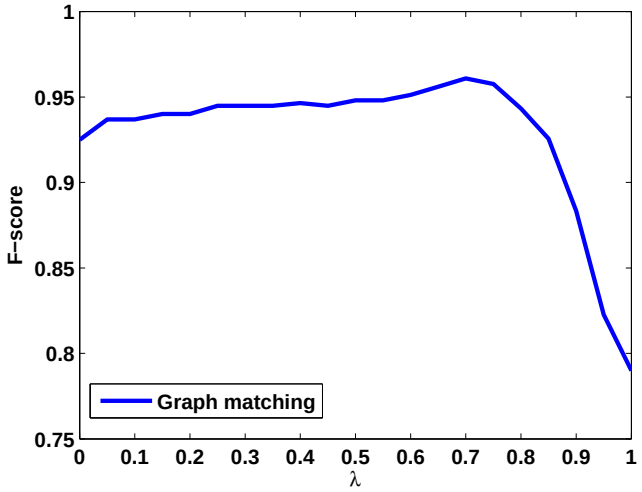


Figure 6.4: Contribution of vertices and edges in the graph matching, which is controlled by λ .

6.3.2 Results on Historical Census Datasets

Finally, we trained the graph model on the whole labelled data set, and applied it to all six historical census datasets. Similar to the experimental settings in [134] and Chapter 5, we classified all household and record links from any pair of consecutive census datasets, e.g. 1851 with 1861, 1861 with 1871, and so on. The matching results are displayed in Table 6.3 for the number of total household matches found on different datasets that include multiple matches of a household in another dataset, and in Table 6.4 for the number of unique household matches for which a household in one dataset is only matched to one household in another dataset. From the tables, it can be observed that both the graph-based methods and the group linking method have generated a much smaller number of total matches and unique matches than the record similarity based methods. Note that the difference between total matches and unique matches are the duplicate matches which are the number of matches where one record in a dataset is matched to two and more records in another dataset. The results indicate that the proposed graph matching methods are very effective reduce number of duplicate matches.

Census dataset pairs	1851–1861	1861–1871	1871–1881	1881–1891	1891–1901
Highest Similarity	2,289	3,032	3,592	3,845	3,998
Vertex Similarity	2,289	3,032	3,592	3,845	3,998
Group Similarity	1,584	2,272	2,827	2,942	3,136
Graph Matching	1,398	1,988	2,452	2,516	2,772
Group Graph	1,492	2,115	2,685	2,756	2,978

Table 6.3: Number of household pairs classified as matched by different household linking methods on historical census datasets.

Census dataset pairs	1851–1861	1861–1871	1871–1881	1881–1891	1891–1901
Highest Similarity	2,509	3,136	3,708	3,938	4,109
Vertex Similarity	2,478	3,090	3,677	3,922	4,091
Group Similarity	1586	2,275	2,830	2,942	3,155
Graph Matching	1,409	1,995	2,462	2,523	2,784
Group Graph	1,493	2,117	2,688	2,756	2,982

Table 6.4: Total household pairs classified as matched in historical census data linkage using different household linking methods.

6.4 Summary

In this chapter, we have introduced a graph matching method to match households in historical census data across time. The proposed graph model considers not only the record similarity, but also incorporates the household structure into the matching step. The similarity between two graphs is calculated as the sum of the vertex and edge similarity. Experimental results have shown that such structural information is very useful in household matching, and when combined with a group linking method, can generate very reliable linking outcome. This method can also be applied to other group record linking applications, in which records in the same group are related to each other.

Chapter 7

Multiple Instance Learning for Household and Record Matching

In this chapter, we propose a novel method to household record linkage based on multiple instance learning (MIL). Our method treats household links as bags and individual record links as instances, such that household links can be classified under a supervised learning paradigm. We further extend multiple instance learning from bag to instance classification by reconstructing bags from candidate instances. The bag reconstruction is based on the modeling of the distribution of negative instances in the training bags using several strategies. The approach allows the selection of negative instances to be combined with a target instance sample to form a bag. Then bag level classification can be applied to predict the bag label which is also the label of a target instance. The classified bag and instance samples lead to a significant reduction in multiple group links, thereby improving the overall quality of linked data.

7.1 Method Overview

In the previous chapters, we have shown that household information can greatly help to improve the accuracy of household and record linkage. A household link will likely contain several links between individual record pairs for its household members. If two households are matching, at least one of their record links has to be

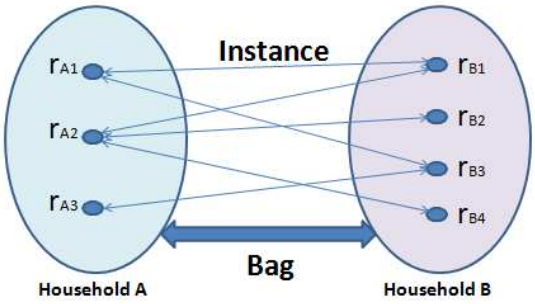


Figure 7.1: An example of group (household) record linkage, and the corresponding MIL setting. Household links are treated as bags, and record links become the instances in the bag.

a match[1]. On the contrary, if two households are not matching, none of their record links shall be matched. This is a typical multiple instance learning (MIL) setting.

MIL is a supervised learning method proposed by Dietterich et al. [106]. In MIL, data are represented as bags, each of which contains some instances. In a binary classification setting, a positive bag contains both positive and negative instances, while a negative bag only consists of negative instances. In the training stage, the class labels are only available at the bag-level but not at the instance-level. The goal of MIL is to learn a classifier which can predict the label of an unseen bag. When applying MIL to the group record linkage problem, group links are treated as bags, and record links become the instances in these bags. A model can then be learned to classify a group link as a match or non-match. Figure 7.1 shows an example of group linking and its relationship to the MIL setting.

Although MIL can be used to classify household pairs as matches or non-matches, the instance classification problem has not been adequately addressed [135]. In traditional MIL methods [110, 111], only the optimal positive instances can be explored in the instance selection step, whilst no explicit instance classification solution has been given. Therefore, there is a gap between MIL and its application to group record

[1]In two matched bags, not all records have to be matched.

linkage. In our work, We extend existing MIL methods to instance level classification by grouping negative instances from the training set with an instance to be classified. This transforms a instance into a bag. We can then employ the bag-level classification model for explicit instance classification. We show that this method can effectively classify both household and record links.

7.2 Multiple Instance Learning

In this section, we introduce the Multiple Instance Learning with Instance Selection (MILIS) method [111], which is an extension of the Multiple-Instance Learning via Embedded Instance Selection (MILES) method [110]. MILES and MILIS are discriminative methods. Compared with other MIL methods as introduced in Section 2.3.4.2, they can easily convert bags into feature vectors via an embedding process so that standard supervised learning approches can be applied. Compared with MILES, MILIS provides a more efficiency and greatly reduce training process speed without compromising the performance [111]. After bag classification, we show how these two methods can be used for bag-level classification.

7.2.1 Definition

In our setting, a bag refers to a household link and an instance refers to a record link. To commence, we give formal definitions of the notion used in the method. Let $\mathcal{B}^+ = \{B_1^+, \ldots, B_{b^+}^+\}$ be a set of positive bags, $\mathcal{B}^- = \{B_1^-, \ldots, B_{b^-}^-\}$ be a set of negative bags, and $b = b^+ + b^-$ be the total number of bags in the training set. A bag B_i contains m_i instances denoted by $\mathbf{x}_{i,j}$ for $j = 1, \ldots, m_i$, with the value for m_i varying from bag to bag. Please note that for our work, the bag refers to household links and instances refer to record links. Each instance $\mathbf{x}_{i,j}$ is associated with a label $y_{i,j} \in \{1, -1\}$ that is not directly observable in the MIL setting, with $y_{i,j} = 1$ corresponding to a match and $y_{i,j} = -1$ to a non-match. The purpose is, therefore, to

predict the binary label value $y_i \in \{1, -1\}$ for a novel test bag $B_i = \{\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,k}\}$, and $y_{i,j}$ for an instance $\mathbf{x}_{i,j}$ in this bag.

7.2.2 Instance Selection and Classifier Learning

Following the idea of instance-based embedding in [110] and instance prototype selection in [111], we generate bag-level feature representations using the similarity between a bag and an instance. Such similarity is based on the Hausdorff distance between a bag and an instance

$$d_H(B_i, \mathbf{x}) = min||\mathbf{x}_{i,j} - \mathbf{x}||^2 \tag{7.1}$$

This distance suggests that a positive instance $\mathbf{x}$ should have a small distance to a positive bag. On the contrary, $\mathbf{x}$ shall have a large distance to a negative bag because it only contains negative instances that are far way from $\mathbf{x}$. An exponential function can be used to convert the distance to similarity as follows

$$s(B_i, \mathbf{x}) = \max_{\mathbf{x}_{i,j} \in B_i} \exp\left(-\gamma||\mathbf{x}_{i,j} - \mathbf{x}||^2\right), \tag{7.2}$$

where γ is a feature mapping parameter that controls the similarity. A bag can now be represented as an n-dimensional vector using the following embedding method

$$z_i = [s(B_i, \mathbf{x}_1^*), \ldots, s(B_i, \mathbf{x}_i^*), \ldots, s(B_i, \mathbf{x}_n^*)], \tag{7.3}$$

where $\mathbf{x}_i^*$ are the prototype instances selected from the training set (as detailed below). This is an improved version of the embedding step in the MILES method [110] which uses all instances in the training set to generate feature vectors.

As proposed in [111], instance prototypes can be generated by selecting the least negative instance from each positive bag and the most negative instance from each

negative bag. This requires modelling of the distribution of negative instances, and computing the probability that an instance has been generated from the negative population. This can be achieved using a kernel density estimation function

$$p(\mathbf{x}|X^-) = \frac{1}{Z\sum_{m_i}} \sum_{j=1}^{m_i} \exp\left(-\beta||\mathbf{x} - \mathbf{x}_j^-||\right), \tag{7.4}$$

where $\mathbf{x}_j^-$ is the j^{th} sample in B^-, Z is a normalisation factor to make $p(\mathbf{x}|X^-)$ a proper density, and β is a parameter to tune the contribution from training samples.

When m_i is very large, the calculation of the kernel density estimation function will be time-consuming. An fast approximation can be implemented by using the k-nearest negative instances from the negative bags B_i^-. The likelihood of $\mathbf{x}$ being negative is then changed to

$$p(\mathbf{x}|B^-) = \frac{1}{Z} \sum_{j=1}^{k} \exp\left(-\beta||\mathbf{x} - \mathbf{x}_j^-||\right), \tag{7.5}$$

where $\mathbf{x}_j^- \in B^-$ is the j^{th} nearest negative neighbour of $\mathbf{x}$. We then select the instance with the lowest likelihood value from each positive bag as the positive instance prototypes (PIPs), and the instance with the highest likelihood value from each negative bag as negative instance prototypes (NIPs). These PIPs and NIPs form the set of instance prototypes (IPs) used in the feature mapping. Using Equations 7.3 and 7.5, we can represent bags in the training set in vector form, and then train an SVM classifier by solving the following unconstrained optimisation problem

$$\min_{\mathbf{w}} \frac{||\mathbf{w}||^2}{2} + C\sum_i \max(1 - y_i(\mathbf{w}^T\mathbf{z}_i), 0), \tag{7.6}$$

where $y_i \in \{1, -1\}$ is the label for bag i, $\mathbf{w}$ is a set of parameters that define a separating hyper-plane, $\mathbf{z}_i$ is the embedded vector for bag i, and C is the regularisation parameter [96].

When a new bag is to be classified, it is first converted into a vectorised representation following Equation 7.3, then the trained SVM classifier can be used for prediction.

7.3 Bag Reconstruction for Instance Classification

In the bag reconstruction step, we applied five different methods that create a group of negative instances with a single instance to create a new bag. Then the bag can be classified using the learned bag-level classifier.

Both MILES and MILIS can find the most positive instance in a positive bag. This is achieved by selecting an instance in the bag that has the lowest likelihood value using Equation 7.5, because a positive bag should contain at least one positive instance. However, when it comes to the situation where a bag contains more than one positive instance, neither method provides an explicit solution to finding all the positive instances. Although a threshold may be set for decision, with instances whose likelihood is higher than the threshold classified as positive, and those with lower as negatives, it is practically difficult to find an appropriate threshold.

Here we propose a method for instance classification by bag reconstruction. We treat each instance in a positive bag as a seed, and group this instance with negative instances to create new bags. Then we apply the trained bag-level classifier to these new bags. If a new bag is classified as positive, then the seed instance is classified as positive. Otherwise, it is classified as negative. This method is based on the fact that if a seed is negative, the reconstructed bag consists of negative instances only, and thus will be classified as negative. Otherwise, the new bag contains one positive instance, therefore, is very likely to be classified as positive.

We commence by a formal definition of the problem. Given training sets $\{\mathcal{B}^+, \mathcal{B}^-\}$ and the learned bag classification model Φ, the goal of instance classification is to pre-

dict the binary label $y_{i,j} \in \{1, -1\}$ of $\mathbf{x}_{i,j} \in B_i$, after B_i has been predicted as positive. In bag reconstruction, $\mathbf{x}_{i,j}$ is grouped with the selected instances $\mathcal{X} = [\mathbf{x}_1^*, \mathbf{x}_2^*, ..., \mathbf{x}_n^*]$, in $\mathcal{B}^-$ to create a new bag $\tilde{B}$. Then the classification model Φ can be used to classify $\tilde{B}$, whose result is also considered as the label for $\mathbf{x}_{i,j}$. The goal of our method is to find the most representative $\mathcal{X}$.

We have proposed five strategies for the bag reconstruction, i.e, **Random**, **K-means**, **Greedy**, **KDE (Kernel Density Estimation)**, and **Kmeans+KDE**, to cope with multiple positive instances in a candidate bag.

7.3.1 Basic Bag Reconstruction Methods

In this section, we introduce three basic bag reconstruction methods to fulfill the task of instance classification for instances in positive bags.

- **Random**

  The random strategy randomly selects negative instances from the training set and groups them with the seed. Therefore, both random negative instances from the training set and the seed instance contribute to the embedding step in MIL.

- **K-means**

  The k-means strategy clusters the NIPs, and uses the cluster centers using the k-means clustering method as the negative instances for bag reconstruction. In this way, we can better represent the distribution of negative instances. Because the number of IPs equals the number of negative bags, which is much smaller than the total number of negative instances in the training set, the clustering can be completed very quickly.

- **Greedy**

  The greedy algorithm is built on top of the random option. With randomly

selected negative instances, a greedy algorithm is adopted which reconstructs new bags and predicts the label of the newly added instance simultaneously.

Here, we give more details on the Greedy strategy. Greedy is built on top of the random option. With randomly selected negative instances, a greedy algorithm is adopted which reconstructs new bags and predicts the label of the newly added instance simultaneously. This guarantees not only the seed, but also the negative instances in the candidate bag contribute to the embedding step. For each instance $\mathbf{x}_{i,j}$ in the candidate bag B_i, we compute its Hausdorff distance to an initial bag $\tilde{B}$ that contains NIPs $\mathbf{x}_k^{*-}$ only:

$$d(\tilde{B}, \mathbf{x}_{i,j}) = \min_{\mathbf{x}_k^{*-} \in \tilde{B}} ||\mathbf{x}_{i,j} - \mathbf{x}_k^{*-}||^2 \tag{7.7}$$

Using this distance measure, we can get the similarity between instance $\mathbf{x}_{i,j}$ and the negative instances in $\tilde{B}$. By ranking the distances of all $\mathbf{x}_{i,j}$ in B_i, we can construct a new bag by sequentially adding into the bag $\tilde{B}$ an instance with the lowest distance among the rest of the instances in the candidate bag B_i. Evaluating the new bag using the bag-level SVM classifier, we can get the label of the newly added instance by taking the bag label. Initially, for a candidate bag that contains both positive and negative instances, the added instances are likely to be negative. Thus, the constructed bag is predicted as negative. When the prediction becomes positive after a new instance is added, the new instance is classified as positive because all other instances in the reconstructed bag are negative. We then replace the positive instance with an instance that has the next larger distance in the rest of the instances in the candidate bag, and re-evaluate the newly reconstructed bag. This process continues until all instances in the candidate bag have been traversed. This strategy is summarised in Algorithm 7.1.

Algorithm 7.1: Instance Classification using Greedy Bag Reconstruction

Input:
- A set $\mathcal{B}^-$ containing all negative bags in the training set
- A bag G containing all NIPs
- A candidate bag B_i that contains m_i instances $\mathbf{x}_{i,j}$ for $j = 1, \ldots, m_i$
- Trained bag-level SVM model Φ
- An empty bag $\tilde{B}$

Output:
- Labels $y_{i,j} \in \{1, -1\}$ for instances $\mathbf{x}_{i,j} \in B_i$, for $j = 1, \ldots, m_i$

1: Randomly sample negative instances from $\mathcal{B}^-$, and add them into $\tilde{B}$
2: **For** $\mathbf{x}_{i,j} \in B_i$ **do**
3: Compute Hausdorff distance $d(\tilde{B}, \mathbf{x}_{i,j})$ using Equation 7.7
4: Sort $d(\tilde{B}, \mathbf{x}_{i,j})$ for $j = 1, \ldots, m_i$
5: **While** B_i is not empty **do**
6. Find $\mathbf{x}_{i,j}$ with the minimum $d(\tilde{B}, \mathbf{x}_{i,j})$ in B_i
7: Add $\mathbf{x}_{i,j}$ into $\tilde{B}$. Remove $\mathbf{x}_{i,j}$ from B_i
8: Classify $\tilde{B}$ using Φ
9: **If** $\tilde{B}$ is negative **then**
10: $y_{i,j} = -1$
11: **Else**
12: $y_{i,j} = 1$. Remove $\mathbf{x}_{i,j}$ from $\tilde{B}$
13: **Goto** step 5

Figure 7.2: Algorithm-Greedy

7.3.2 Bag Reconstruction by Kernel Density Estimation

The classification performance of the bag reconstruction method is dependent on the quality of the selected negative instances selected for reconstruction. One would expect that these new bags shall be consistent with the distribution of the bags in the training set, so that the learned classification model Φ will generate good classification results. However, the random and greedy instance selection strategies have not taken the data distribution into consideration. This means the quality of the bag reconstruction is not guaranteed due to the uncertainty in the negative instance selection. To solve this problem, we seek to model the distribution of the instances in the negative training bags and propose a new method to improve the bag reconstruction method.

Note that Equation 7.5 defines a kernel density estimator with an isotropic[2] Gaussian kernel [111]. This allows the modeling of the likelihood that an instance $\mathbf{x}$ is contained in the negative bags. Based on this observation, our first solution is to select the most negative instance in the negative bags as the member of $\mathcal{X}$. Thus, $\mathbf{x}^*$ is defined by

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{B}^-} p(\mathbf{x}|\mathcal{B}^-) \tag{7.8}$$

where $p(\mathbf{x}|\mathcal{B}^-)$ is given by Equation 7.5.

This solution is similar to the MILIS negative IP selection process. The difference lies in that an IP is selected from a single bag in MILIS, while the $\mathbf{x}^*$ is selected from the whole negative instance pool. Such an option has three advantages. Firstly, from the data distribution point of view, $\mathbf{x}^*$ will be close to the NIPs and far away from the PIPs. Because the bag level feature representation step is performed using Equation 7.2, the similarity between a bag and an instance prototype is based on the instance in the bag that is most similar to the instance prototype. Thus, with the most negative instance being selected as $\mathbf{x}^*$, it is guaranteed that high similarity to NIPs can be achieved. Secondly, the selection of $\mathbf{x}^*$ is deterministic. Unlike the random selection strategy, the most negative instance in the negative training bags is unique. Thirdly, the reconstruction of all instances to be tested uses the same $\mathbf{x}^*$, which is not dependent on the testing data or the number of iterations to be performed as in the k-means and greedy strategies described before. Therefore, this approach is very efficient. A summary of this instance classification method is given in Algorithm 7.2.

When the data are generated from a mixture of Gaussian models or from an arbitrary distribution, it may be necessary to select multiple instances for bag reconstruction. Therefore, $\mathbf{x}^*$ is expanded to a set of instances $\mathcal{X} = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_k^*\}$. This leads to a larger reconstructed bag. A simple method of generating such an $\mathcal{X}$ is to iteratively

[2]Isotropy means uniformity in all orientations.

Algorithm 7.2: Instance Classification using KDE Bag Reconstruction

Input:
- Training set $\mathcal{B} = \{\mathcal{B}^+, \mathcal{B}^-\}$
- A testing bag B_i that contains m_i instances $\mathbf{x}_{i,j}$ for $j = 1, \ldots, m_i$

Output:
- Label $Y_i \in \{1, -1\}$ for bag B_i and labels $y_{i,j} \in \{1, -1\}$ for instances $\mathbf{x}_{i,j} \in B_i$

1. Generate IPs using Equation 7.5 and MILIS instance selection strategy
2. Calculate instance-based embedding for bag feature representation using Equation (7.3)
3. Train bag-level SVM model Φ
4. Classify B_i
5 **If** $Y_i = -1$ **then**
6. Classify all $\mathbf{x}_{i,j} \in B_i$ as negative
7. **Else**
6. Create $\mathcal{X}$ based on the distribution of negative training bags
8. **For** $\mathbf{x}_{i,j} \in B_i$ **do**
9. Create a reconstructed bag $\tilde{B} = \{\mathbf{x}_{i,j}, \mathcal{X}\}$
10. Classify $\tilde{B}$ using Φ
11. **If** $\tilde{B}$ is negative **then**
12. $y_{i,j} = -1$
13. **Else**
14. $y_{i,j} = 1$

Figure 7.3: Algorithm-Kernel Density Estimation

search for $\mathbf{x}^*$ from the remaining negative instances in $\mathcal{B}^-$ without replacement. This guarantees the retrieval of the most negative instances based on kernel density estimation. However, there is a high possibility that several selected negative instances are very close to each other. Then the contributions of these instances to the instance embedding step are similar. This means that the $\mathcal{X}$ may contain redundant information. On the other hand, some important negative instances may be missed.

This problem can be illustrated by an example shown in Fig. 7.4. In this example, the data are generated from the sum of six Gaussian distributions with means -2.1, -1.3, -0.4, 1.9, 5.1, and 6.2, respectively. The standard deviation of the Gaussian distribution is set to 1. It can be seen that there are two peaks in the curve. When $\mathcal{X}$ contains only a single element, $\mathbf{x}_1^*$ will be selected due to the highest probability

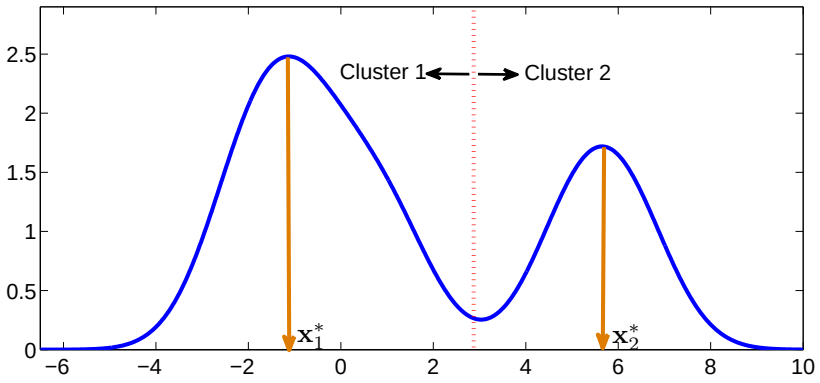


Figure 7.4: Kernel density estimation for instance selection in bag reconstruction.

density at the location. If more than one element in $\mathcal{X}$ is needed, it is most likely that points surrounding $\mathbf{x}_1^*$, will be selected, while $\mathbf{x}_2^*$ is missed.

To solve this problem, we introduce the second solution, which is based on dividing the feature space of negative instances into subspaces, and then applying kernel density estimation on each subspace. The subspace division can be performed by k-means clustering, which partitions the $\mathcal{B}^-$ into K sets $\mathcal{B}^- = \{\mathcal{B}_1^-, \ldots, \mathcal{B}_K^-\}$. For each set $\mathcal{B}_k^-$, we run kernel density estimation on all the negative instances in it. Therefore, Equation 7.5 is modified as

$$p(\mathbf{x}_{i,j}|\mathcal{B}_k^-) = \frac{1}{Z}\sum_{t=1}^{T}\exp\left(-\beta||\mathbf{x}_{i,j} - \mathbf{x}_t^-||\right), \tag{7.9}$$

where T is the total number of negative instances in $\mathcal{B}_k^-$. The negative instance selection rule in Equation 7.8 is updated correspondingly as

$$\mathbf{x}_k^* = \arg\max_{\mathbf{x}\in\mathcal{B}_k^-} p(\mathbf{x}_{i,j}|\mathcal{B}_k^-) \tag{7.10}$$

This allows both $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ in the above example be selected, which are the most representative instances. Note that when $K = 1$, Equations 7.9 and 7.10 reduce to the single element case in Equations 7.5 and 7.8.

7.4 Experiments and Evaluation

We have conducted experiments on both synthetic data and labeled historical census data to evaluate the effectiveness of our household and instance classification methods. In bag level classification, we use two multiple instance learning methods, MILES and MILIS, to verify the bag classification accuracy. For the implementation of MILES, we have used the MOSEK[3] system to solve the linear programming formulation in the one-norm SVMs. To train the MILIS algorithm, we have used LIBLINEAR [136]. The SVM regularisation parameter C was set using grid search on the training data. For Equation 7.5, we set $K = 10$ which is the same as in [111]. The feature mapping parameter γ in Equation 7.2 and the scale parameter β for the likelihood estimation in Equation 7.5 are set to 1 and 10, respectively, according to [111]. For bag reconstruction in instance classification for the census data experiments, we have grouped a seed with five random negative instances. This is based on the fact that by average, a bag in the census datasets contains 5.65 instances, as can be calculated from Table 7.2.

7.4.1 Household Classification Results

We have randomly selected 1,250 positive household links and 1,250 negative household links from the 1871 and 1881 datasets. The method to generate these negative household links has been introduced in Chapter 5. To show the performance of the MILES and MILIS methods on household link classification, we performed 10-fold cross validation on the randomly split labelled data, with half used for training and half for testing. Thus, each training set and testing set contains 625 positive and 625 negative household pairs.

Both the MILES and MILIS methods show similar and high accuracy, achieving $88.37 \pm 1.13\%$ and $92.02 \pm 1.29\%$ accuracy on household link classification, respec-

[3] http://www.mosek.com

Table 7.1: Number of positive bags detected in different pairs of historical census datasets using the MILES and MILIS methods.

Census dataset pairs	1851–1861	1861–1871	1871–1881	1881–1891	1891–1901
MILES-bag	4,285	5,626	6,322	6,112	6,173
MILIS-bag	6,383	7,891	8,802	8,323	7,792

Table 7.2: Number of bags and instances generated from historical census data pairwise linking results. Threshold ρ is set to 5 out of 10.

Census dataset pairs	1851–1861	1861–1871	1871–1881	1881–1891	1891–1901
Number of instances	664,793	884,473	954,260	921,453	1,042,592
Number of bags	266,470	370,787	394,840	380,372	435,291

tively. This shows the effectiveness of applying multiple instance learning methods for the household linking problem. When efficiency is considered, MILIS shows superior performance than MILES. The MILES method took 48.25 ± 3.53 seconds for training, and 1.82 ± 0.08 seconds for testing, while MILIS only took 9.83 ± 0.57 for training and 1.09 ± 0.23 seconds for testing.

In the next experiment, we re-trained the MILES and MILIS models using all the labelled data, and then classified all household links from any pair of consecutive census datasets, e.g. 1851 with 1861, 1861 with 1871, and so on. The results are shown in Table 7.1. As shown in this table, MILES and MILIS showed mixed performance on the bag-level classification, each having generated more positive bags than the counterpart on some datasets. By comparing the number of matched households with the total number of households in each census dataset from Table 4.3, one can observe that the results contain multiple matches. This is expected because of two reasons. First, a household may split into several households, for example, due to the moving out of grown-up children, or two households might merge when widowed individuals form a new household. Second, there are many similar record pairs among different households, which may have generated false positive results.

7.4.2 Instance Classification Results

Now we turn our attention to instance classification results. For bag reconstruction of instance classification, we have grouped each seed (instance) in a positive bag with several negative instances, and then applied bag level classification models introduced in previous sections to classify these instances. To compare different bag reconstruction strategies, we have performed experiments on synthetic data, labelled household and record pairs, and the whole historical census datasets. The synthetic data allows theocratical analysis of the robustness of our proposed bag reconstruction methods. Experiments on the labeled data gives quantitative evaluation of different methods. Finally, we compare the performance of various methods on the historical census data.

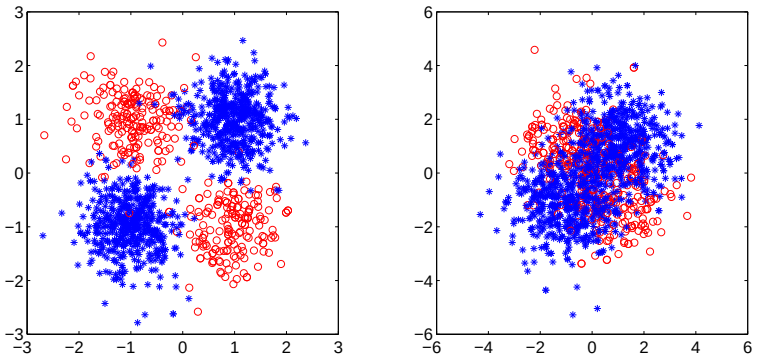


Figure 7.5: Two examples of synthetic data set. Each dataset is generated with four Gaussian distributions, whose means are $[1, 1]$, $[-1, 1]$, $[1, -1]$, and $[-1, -1]$, respectively. The standard deviation of the Gaussian distribution in the left panel is set to 0.5. The standard deviation of the Gaussian distribution in the right panel is set to 1.

7.4.3 Results on Synthetic Data

We created synthetic data to analyse the behaviour of our methods. Two examples of the synthetic datasets are shown in Fig. 7.5. Each dataset contains 1,000 positive instances (red circles) and 5,000 negative instances (blue asteroids). These instances were randomly generated from four Gaussian distributions, two for positive

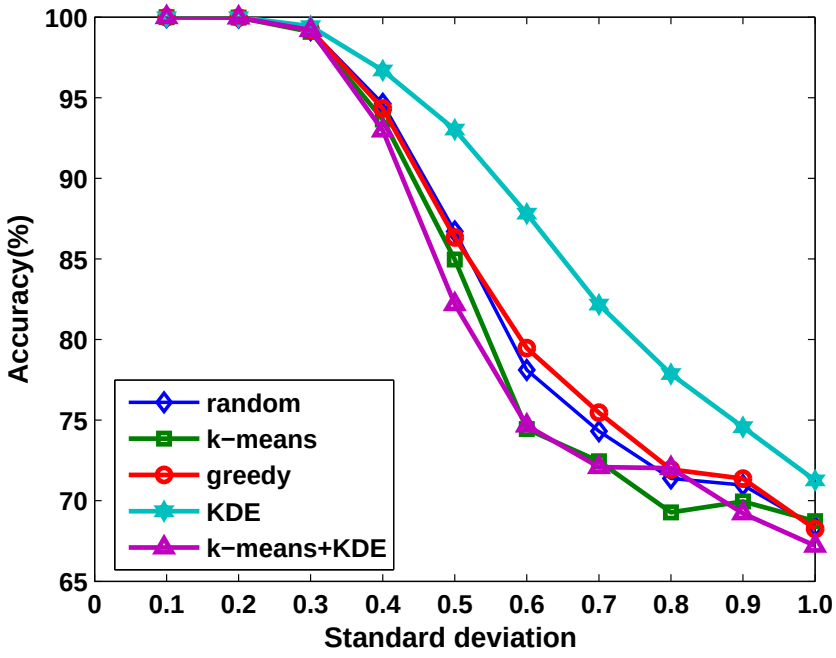


Figure 7.6: Comparison of instance classification accuracies when MILIS is combined with different bag reconstruction methods. The data were generated with different levels of difficulties controlled by the standard deviations of Gaussian distribution.

instances and two for negative instances. The means of the Gaussian distributions used to generate these two datasets are identical, but their standard deviations are different. With larger standard deviation, the positive and negative instances are more overlapping with each other, and thus are more difficult to be classified. We constructed positive bags by randomly sampling from both positive and negative instances. Negative bags were constructed in a similar manner, but only from negative instances. Each bag contains a random number of instances ranging from 1 to 10. In this way, we have generated 1,000 positive bags and 1,000 negative bags. In the experiments, the instance labels are only used for evaluation purpose, without being accessed in the training stage.

In the first experiment, we compared the robustness of the proposed bag reconstruction methods when the difficulty level of data varies. We have randomly divided the

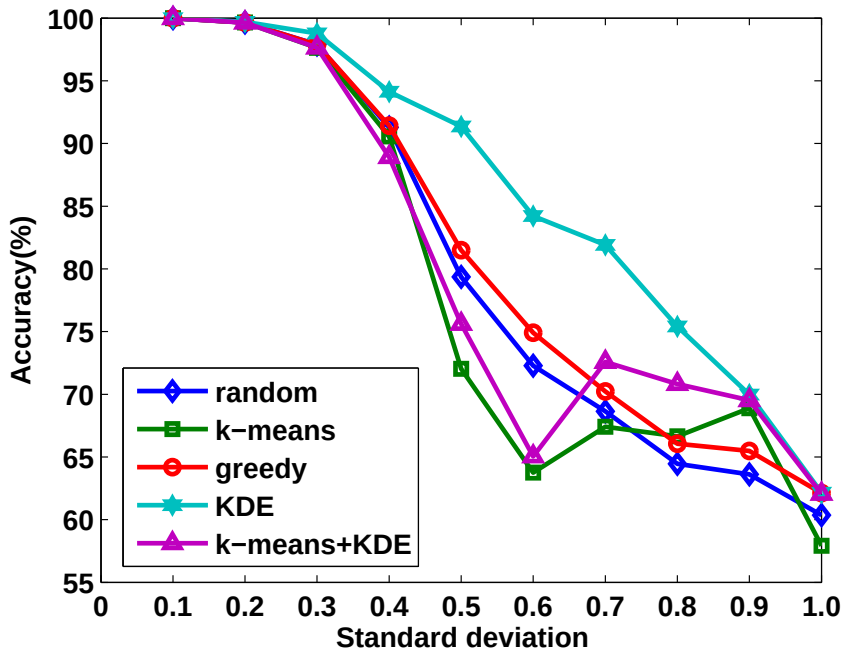


Figure 7.7: Comparison of instance classification accuracies when MILES is combined with different bag reconstruction methods. The data were generated with different levels of difficulties controlled by the standard deviations of Gaussian distribution.

synthetic data into a training set and a testing set with equal number of positive and negative bags. The bag level classifier was learned from the training set using the MILIS algorithm [111]. The instance level classification is only applied to positive testing bags because we already know that the negative bags contain only negative instances from the bag definition, while positive bags contain both positive and negative instances. Those positive instances correspond to the matched records that we want to identify. To achieve this goal, we applied different bag reconstruction methods for instance classification.

In this experiment, the number of instances selected for bag reconstruction is set to 5, which is the average number of instances in synthetic bags. The standard deviations of the Gaussian distributions are set from 0.1 to 1.0 with 0.1 in interval. The experiments are run for 10 times, with randomly split training and testing sets. Fig. 7.6

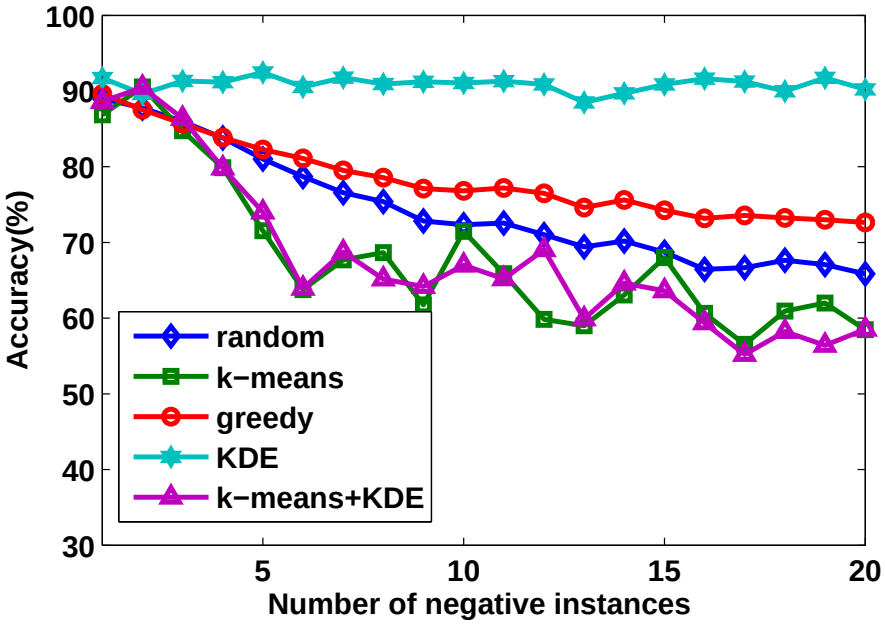


Figure 7.8: Influence of number of negative instances used for bag reconstruction, when MILES is used with different bag reconstruction methods.

and Fig. 7.7 displays the mean accuracy of each method. The results show that when the difficulty of the data is low, all methods perform similarly well. However, after the standard deviation of Gaussian distribution is set to a value larger than 0.3, the proposed KDE method achieves much higher accuracy than the alternatives. This implies that using the most negative instances in the negative training bags for bag reconstruction is the most reliable approach among all methods being compared. On the other hand, the alternative methods do not show much differences in their performance.

In the second experiment, we analyzed the influence of the number of negative instances in $\mathcal{X}$, i.e, the size of the reconstructed bags, on the bag reconstruction when MILES and MILIS are combined with different bag reconstruction methods. Here, the standard deviation of Gaussian distribution for data generation is set to 0.5 for moderate level of data difficulty. As shown in Fig. 7.8 (for MILES) and Fig. 7.9 (for MILIS), the KDE method performs the best among all instance classification meth-

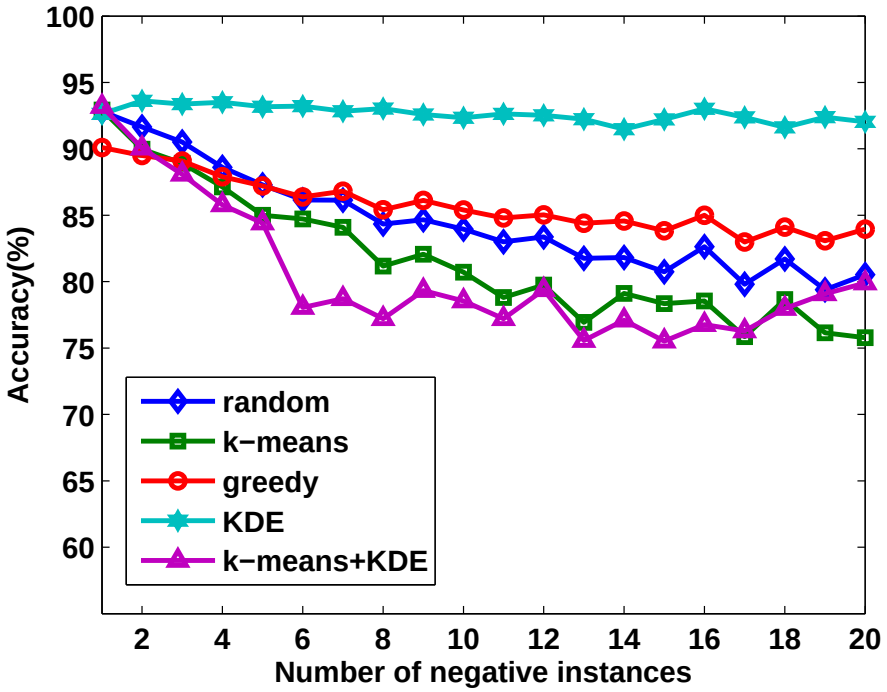


Figure 7.9: Influence of number of negative instances used for bag reconstruction, when MILIS is used with different bag reconstruction methods.

ods in overall performance with a very stable accuracy curve. The highest accuracy is achieved at 5 negative instances, which is only slightly higher than the accuracy with 1 negative instance. This is reasonable because the selected most negative instances may be close to each other as described in Section 7.3.2. Therefore, the contribution of these instances, no matter how many they are, are similar in the feature embedding step of the reconstructed bags. This implies that the reconstruction with the most negative instance in the training set is sufficient to achieve good performance.

This observation can greatly simplify the gene $\mathcal{X}$ because now we only need to find the most negative instance in the negative training bags, which is already available from the MILIS IP generation step. On the other hand, the option of instance selection from clustered negative instances is not very stable due to the randomness of the initialization of the k-means clustering method. The accuracy of other methods

	Instance Classification (%)				
MILIS	Random	K-means	Greedy	KDE	Kmeans+KDE
Mean	79.74	78.22	74.93	84.59	82.25
Standard Deviation	3.56	10.28	3.45	4.48	9.48
MILES	Random	K-means	Greedy	KDE	Kmeans+KDE
Mean	88.23	89.68	86.35	91.20	84.18
Standard Deviation	1.40	4.13	2.85	1.66	9.57

Table 7.3: MILIS Instance classification on labeled data.

under comparison is greatly affected by the number of instances used for bag reconstruction. All of these methods achieve the highest accuracy with 1 negative instance. When the bag size increases, their accuracies drop significantly.

7.4.4 Results on Historical Census Data

Now we show the performance of bag reconstruction methods on historical census data. Similar when in evaluating the group linking methods in Chapter 5, in the first experiment, we use our labeled data to give a quantitative evaluation of the methods under comparison. We split the 2,500 manually labeled positive household links randomly into a training and a testing set, then run the experiments for 10 times to obtain average mean and standed deviation values. All the bag reconstruction results using the MILES and MILIS models are shown in Table 7.3. The results show that the proposed KDE method can achieve the highest accuracy. At the same time, the Random and the Greedy methods also have achieved very high classification performance. The k-means clustering based methods, however, have demonstrated relatively low performance compared with the other options. In particular, the standard deviations of these two methods are very high, which suggests the performance of clustering methods is closely related to the quality of the random initialisation used.

In the next experiment, we re-trained the MILES and MILIS models using all labelled data, and then classified all household and record links from all consecutive census datasets. The bag classification leads to the generation of positive and negative bags,

which correspond to matched and non-matched household links. Based on these results, we performed instance classification. Because negative bags contain only negative instances, only positive classified bags need to be investigated. To do so, we ran the five bag reconstruction methods as introduced in Section 7.3.2 to convert each instance in the positive bags to the reconstructed bags, then applied both MILES and MILIS models to classify these bags. The results are shown in Table 7.4. For instance-level classification, it can be seen that all methods have generated similar numbers of positive instances. It is difficult to evaluate the accuracy of this classification because we don't have the ground truth. In order to produce more reliable results, we performed result fusion by selecting those common positive instances predicted by all five bag reconstruction methods. These are the most consistent record links for each pair of census datasets.

The number of household matches after this fusion process for both MILIS and MILES bag reconstruction methods are also presented in Table 7.4. The results show that MILIS-based methods have generated more matched record pairs than the MILES-based methods. This is natural because MILIS has generated more positive bags than MILES in household link classification as shown in Table 7.1.

Census dataset pairs	1851–1861	1861–1871	1871–1881	1881–1891	1891–1901
MILIS-random-instance	7,600	9,921	11,898	11,960	11,532
MILIS-kmeans-instance	6,747	8,842	10,932	11,905	10,834
MILIS-greedy-instance	6,782	8,799	10,803	10,804	10,369
MILIS-KDE-instance	6,757	8,869	10,938	11,102	10,925
MILIS-kmeansKDE-instance	6,734	8,828	10,897	11,050	10,881
After result fusion	5,115	6,879	8,500	8,742	8,593
MILES-random-instance	4,880	6,638	8,100	7,667	8,284
MILES-kmeans-instance	7,368	9,810	11,657	11,120	11,403
MILES-greedy-instance	5,297	7,051	8,479	8,204	8,730
MILES-KDE-instance	5,930	8,086	9,804	9,612	10,006
MILES-kmeansKDE-instance	6,802	9,203	11,161	10,736	10,816
After result fusion	4,202	5,809	7,087	6,830	7,363

Table 7.4: Number of positive instances detected in different pairs of historical census datasets for each method under comparison. This table also includes the outcome of result fusion.

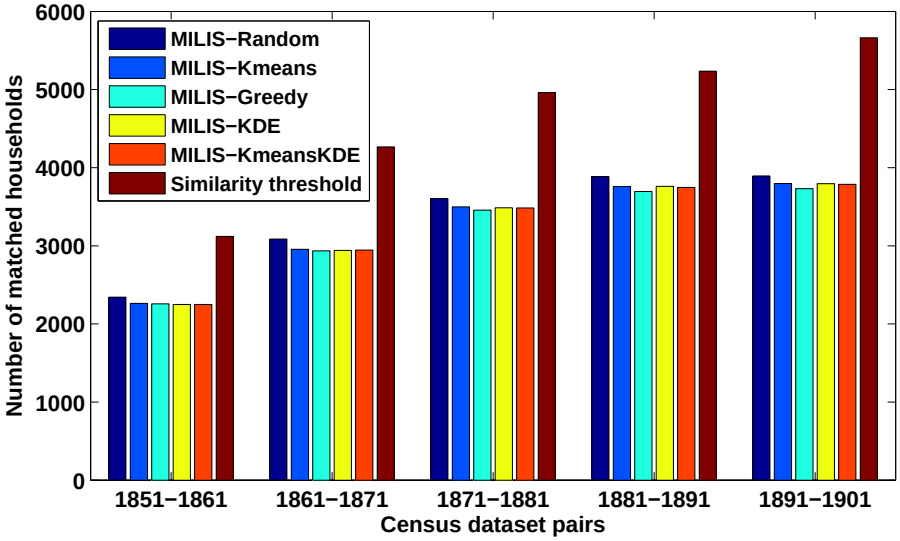


Figure 7.10: Household matching results after group linkage step.

From Table 7.1, it also can be seen that the numbers of household matches found in each pair of census datasets are higher than the number of households in the original datasets as presented in Table 4.3. This implies that some households have multiple matches. To reduce the number of multiple matches, we need to apply the group linking method introduced in Chapter 5. Figure 7.10 shows the results of the MILIS-based methods and the similarity threshold method after group linking has been applied [102]. The results show that different bag reconstruction methods have similar performance, but the similarity threshold method has generated many more matches than the MILIS-based methods.

7.5 Summary

In this chapter, we have introduced a group record linkage method based on multiple instance learning (MIL), and evaluated this method on real historical census data. In this method, groups of links are considered as bags and associated record links are treated as instances, with only the bag-level labels provided. The multiple instance learning paradigm has provided the group linkage problem with a suitable

supervised learning tool to classify groups, even if the labels of record links are not available. We compared the MILIS method with the MILES method in bag level classification. The experiments have shown that these two methods are effective on historical census data.

Besides household classification, we also presented a novel method of instance classification by bag reconstruction. This method models the distribution of the negative training bags, and groups the most representative negative instances with the target instance to be classified in order to covert instance classification to bag classification. Experimental results show that this method is very effective, and has outperformed several baseline methods based on random and clustering sampling strategies. Analysis on the results also suggests that very few instances are required for the bag reconstruction purpose, which allows fast and convenient bag reconstruction.

Chapter 8

Conclusions and Future Work

8.1 Summary of the Thesis

In this thesis, we have provided a systematic solution to the problem of historical census data linkage. Our goal is to automatically link individual records and households in historical census data, so as to relieve social scientists from tedious manual data linking tasks, and provide them with new tools for historical census data analysis. Our solution has covered several key steps in the data linkage process, including data cleaning and standardisation, pair-wise comparison, and data linkage classification, with special focus on the classification step. Different from traditional record linkage approaches that only perform linkage at the record pair level, we adopt a strategy to consider a household as an integrated entity, and use whole of household information to improve the effectiveness of data linkage. Our research has lead to the development of three record and household linkage methods.

The first method is based on group linking. It first computes attribute similarities between record pairs and uses these similarities as the input to a support vector machine classifier which classifies record pairs into a matched and a non-matched class. The classification outcome forms the input to the household linking step. A group linking technique is then used to generate household linking similarities. The Jaccard and Bipartite similarities measures are used in the group linking models, and their performances has been compared. The results show that when combining support vector machine classification for record linking with group linking using the Jaccard

similarities measure, the household linkage approach generates better results than compared to using the alternative methods under comparison.

The second method adopts a graph matching approach to match households across time. The proposed graph model considers not only record similarity, but also incorporates the structural information of households into the matching step. Experimental results have shown that such structural information is very useful in household matching practise, and when combined with a group linking method, can generate very reliable linking outcomes. This method can easily be applied to other group record linking applications, in which records in the same group are related to each other.

The third method uses and extends a multiple instance learning approach. It is an integrated solution for both record and household level link classification. In this method, household links are considered as bags and associated record links are treated as instances, with only the bag-level labels provided. The multiple instance learning paradigm has provided the household linkage problem with a suitable supervised learning tool to classify household links, even if the labels of record links are not available. Once a household link classification model is trained, this method groups the most representative negative instances, which are modelled by the distribution of the instances in the negative training bags, with the target instance (record link) to be classified. This step reconstructs target instances into bags, so that the trained household link classifier can be used to predict whether a record pair is matched or not. Experimental results show that this approach is very stable and effective. In particular, MILIS based methods have shown advantages in both bag and instance classifications compared to the MILE based method, with 3.65% and 6.61% margins over MILES in bag and instance classification at their highest accuracies, respectively. Analysis on the results also suggest that very few instances are

required for the bag reconstruction purpose, which allows fast and convenient bag reconstruction.

8.2 Discussion

The core idea of the group-based linkage methods introduced in our thesis is that linking both individual records and households leads to a reduction of ambiguous links and achieves highly accurate linkage results. The developed methods aim at removing as many multiple links as possible, and they do not reduce the number of records with single links. During the development of these methods, domain knowledge on census data and household information plays an important role, which leads to the improvement of record linking results. The model development also shows that the combination of supervised learning and group data processing methods for historical census household linkage is very effective, which can be exemplified by the experimental results as introduces in Chapters 5-7

We have explored three group-based linkage options. The group linking method reported in Chapter 5 calculates the matched record links out of distinctive members in two households being linked. Such a strategy does not take the distribution of linking data and the relationship between household members into consideration. Therefore, this method does not perform as good as the other two options. However, the advantages of group linking comes from its efficiency, and the fact that it may not need training data to generate a prediction model. These properties are very suitable for linking large-scale data sets, which is prevalent in the Big Data era, when ground truth data are difficult to obtain.

The graph-based method also has its advantage and disadvantage. It considers the household structure, which is valuable information to compare two households, as the relationships within a household are normally the most steady properties of a

household. The experimental results have shown the effectiveness of this approach at improving the linking accuracy. On the other hand, the construction and comparison of graphs is normally a time consuming process, especially when there are large number of members in a household. Therefore, preprocessing steps have to be included to remove household members that do not have a high possibility of matching, i.e., using a threshold on record similarities, in the households to be matched.

Amongst the three models based on our experimental evaluations, the one based on multiple instance learning is the most promising one. This is not only because it provides an integrated solution to both record level and household level linkage, but also because it takes the data distribution at both bag and instance levels into consideration. This allows the generation of a more reliable prediction model that can be generalised to unseen data samples. The current instance reconstruction approach is based on the distribution of negative instances. It is expected that if approaches can be developed to better characterise the data distribution, superior classification performance can be achieved.

8.3 Future Work

Extension of the methods reported in this thesis can be explored in three aspects. Firstly, the existing methods only link two historical census datasets. The linking across time is implemented by propagating and integrating the pair-wise dataset linking results. It is necessary to develop approaches that can link records in multiple datasets simultaneously. This can be achieved by treating the historical census data as time series, and use graphical models to perform forward and backward optimisation on the probability that a record in the first dataset is matched to a certain record in the second dataset. Such methods may also facilitate the generation of family trees, which is of interest to social scientists.

Secondly, we have only explored two multiple instance learning models, i.e., MILES [110] and MILIS [111]. Future work could extend this strategy to other multiple instance learning approaches such as DD-SVM [108] and SMILE [109]. Regarding the bag reconstruction approach, it is necessary to develop models that can automatically determine the number of instance prototypes that are needed for reconstruction, and select those that better describe the data distribution of bags other than all negative instances. Furthermore, it would be interesting to develop iterative instance prototype selection methods, so that the reconstructed bags can be gradually optimised by using the training samples.

Thirdly, the reported group-based methods are general in natural, and can easily be extended to other datasets that require both individual and group level classification, such as bibliographic databases, health record depository, or commercial consumer databases. We will continue exploring possible applications of our methods, and develop semi-automatic approaches to meet the needs of real-world application requirements.

Lastly, our research is under an assumption that a household in one dataset can be matched to at most one household in another dataset. Thus, our purpose is to find households with the majority of their members matched. However, during the interval between two censuses, a household may split into multiple households because children in a household may get married and move out to another household, and servants may change jobs or get married. In the future, we aim to develop methods to cope with such scenario.

Bibliography

1. E. Higgs, "A clearer sense of the census: Victorian censuses and historical research," in *Public Record Office Handbooks*, no. 28. Her Majesty's Stationery Office, London, 1996. 1, 35

2. D. Quass and P. Starkey, "Record linkage for genealogical databases," in *ACM KDD Workshop*, Washington DC, 2003, pp. 40–42. 1, 2, 67

3. S. Ruggles, "Linking historical censuses: a new approach," *History and Computing*, vol. 14, no. 1+2, pp. 213–224, 2006. 1, 2, 32, 33, 67

4. E. Glasson, N. de Klerk, A. Bass, D. Rosman, L. Palmer, and C. Holman, "Cohort profile: The Western Australian family connections genealogical project," *International Journal of Epidemiology*, vol. 37, pp. 30–35, 2008. 1, 2, 67

5. G. Bloothooft, "Multi-source family reconstruction," *History and Computing*, vol. 7, no. 2, pp. 90–103, 1995. 2, 67

6. ——, "Assessment of systems for nominal retrieval and historical record linkage," *Computers and the Humanities*, vol. 32, no. 1, pp. 39–56, 1998. 2, 67

7. E. Fure, "Interactive record linkage: The cumulative construction of life courses," *Demographic Research*, vol. 3, p. 11, 2000. 2, 67

8. A. Reid, R. Davies, and E. Garrett, "Nineteenth century Scottish demography from linked censuses and civil registers: a 'sets of related individuals' approach," *History and Computing*, vol. 14, no. 1+2, pp. 61–86, 2006. 2, 67

9. M. Sadinle and S. Fienberg, "A generalized Fellegi-Sunter framework for multiple record linkage with application to homicide record systems," *Journal of the American Statistical Association*, vol. 108, no. 502, pp. 385–397, 2013. 2, 29

10. Z. Fu, P. Christen, and M. Boot, "A supervised learning and group linking method for historical census household linkage," in *Proceedings of the 19th Ninth Australasian Data Mining Conference*, Ballarat, Australia, 2011. 5, 16, 68

11. I. Bhattacharya and L. Getoor, "Collective entity resolution in relational data," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, p. Article No. 5, 2007. 13, 28, 70

12. L. Getoor and C. P. Diehl, "Link mining: a survey," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005. 13

13. S. Tejada, C. Knoblock, and S. Minton, "Learning object identification rules for information integration," *Information Systems*, vol. 26, no. 8, pp. 607–633, 2001. 13

14. ——, "Learning domain-independent string transformation weights for high accuracy object identification," in *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 350–359. 13

15. M. Scannapieco, I. Figotin, E. Bertino, and A. K. Elmagarmid, "Privacy preserving schema and data matching," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of data*, 2007, pp. 653–664. 13

16. H. L. Dunn, "Record linkage," *American Journal of Public Health*, vol. 36, no. 12, pp. 1412–1416, 1946. 13

17. H. Newcombe, J. Kennedy, S. Axford, and A. James, "Automatic linkage of vital records," *Science*, vol. 130, no. 16, pp. 954–959, 1959. 14

18. M. Odell and R. Russell, "The soundex coding system," US Patents, 1918. 14

19. H. Newcombe, M. Fair, and P. Lalonde, "The use of names for linking personal records," *Journal of the American Statistical Association*, vol. 87, no. Issue 420, pp. 1193–1204, 1992. 14

20. I. P. Fellegi and A. B. Sunter, "A theory of record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969. 14, 17, 29, 97

21. W. E. Winkler, "The state of record linkage and current research problems," US Bureau of the Census, Technical Report, 1999. 15, 25

22. ——, "Methods for record linkage and Bayesian networks," US Bureau of the Census, Technical Report, 2002. 15, 27

23. G. Weber, S. Murphy, A. McMurry, D. Macfadden, D. Nigrin, S. Churchill, and I. Kohane, "Federated queries of clinical data repositories: the sum of the parts does not equal the whole." *Journal of the American Medical Informatics Association*, vol. 20, no. e1, pp. 155–163, 2013. 15

24. R. Bell, J. Keesey, and T. R. T, "The urge to merge: linking vital statistics records and medicaid claims," *Medical Care*, vol. 32, no. 10, pp. 1004–1018, 1994. 15

25. D. Clark, "Practical introduction to record linkage for injury research," *Injury Prevention*, vol. 10, no. 3, pp. 186–191, 2004. 15

26. J. Hurdle, S. Haroldsen, A. Hammer, C. Spigle, A. Fraser, G. Mineau, and S. Courdy, "Identifying clinicaltranslational research cohorts: ascertainment via querying an integrated multi-source database," *Journal of the American Medical Informatics Association*, vol. 20, no. 1, pp. 164–171, 2013. 15

27. M. Kuzu, M. Kantarcioglu, E. A. Durham, C. Toth, and B. Malin, "A practical approach to achieve private medical record linkage in light of public resources," *Journal of the American Medical Informatics Association*, vol. 20, no. 2, pp. 285–292, 2013. 15

28. K. M. Campbell, D. Deck, and A. Krupski, "Record linkage software in the public domain: a comparison of Link Plus, the Link King, and a 'basic' deterministic algorithm," *Health Informatics Journal*, vol. 14, no. 1, pp. 5–15, 2008. 15, 16

29. A. G. Pacheco, V. Saraceni, S. H. Tuboi, L. H. Moulton, R. E. Chaisson, S. C. Cavalcante, B. Durovni, J. C. Faulhaber, J. E. Golub, B. King, M. Schechter, and L. H. Harrison, "Validation of a hierarchical deterministic record-linkage algorithm using data from 2 different cohorts of human immunodeficiency virus-infected persons and mortality databases in Brazil," *Practice of Epidemiology*, vol. 168, no. 11, pp. 1326–1332, 2008. 15

30. S. Grannis, J. Overhage, and C. McDonald, "Analysis of identifier performance using a deterministic linkage algorithm," in *Proceedings of the AMIA Annual Symposium*, 2002, pp. 305–309. 15

31. M. Jaro, "Probabilistic linkage of large public health data files," *Statistics in Medicine*, vol. 15, no. 5-7, pp. 491–498, 1995. 15

32. S. Gomatam, R. Carter, M. Ariet, and G. Mitchell, "An empirical comparison of record linkage procedures," *Statistics in Medicine*, vol. 21, no. 10, pp. 1485–1496, 2002. 15

33. M. Tromp, N. Meray, A. Ravelli, J. Reitsma, and G. Bonsel, "Ignoring dependency between linking variables and its impact on the outcome of probabilistic record linkage studies," *Journal of the American Medical Informatics Association*, vol. 15, no. 5, pp. 654–660, 2008. 15

34. Y. Qu, M. Tan, and M. Kutner, "Random effects models in latent class analysis for evaluating accuracy of diagnostic tests," *Biometrics*, vol. 52, no. 3, pp. 797–810, 1996. 15

35. M. Tromp, A. Ravelli, G. Bonsel, A. Hasman, and J. Reitsma, "Results from simulated data sets: probabilistic record linkage outperforms deterministic record linkage," *Journal of Clinical Epidemiology*, vol. 64, no. 5, pp. 565–572, 2011. 15

36. M. Schraagen, "Historical record linkage using event sequence consistency," in *Proceedings of the Workshop on Population Reconstruction*, Amsterdam, Netherlands, 2014. 15

37. P. Domingos, "Multi-relational record linkage," in *Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining*, 2004, pp. 31–48. 16, 29

38. P. Li, X. L. Dong, A. Manurino, and D. Srivastava, "Linking temporal records," *Frontiers of Computer science*, vol. 6, no. 3, pp. 293–312, 2012. 16

39. A. Arasu, C. Re, and D. Suciu, "Large-scale deduplication with constraints using dedupalog," in *Proceedings of the IEEE 25th International Conference on Data Engineering*, no. 592–563, 2009. 16

40. M. Hernandez and S. Stolfo, "The merge/purge problem for large databases," in *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, 1995, pp. 127–138. 16

41. P. Christen, *Data Matching-Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, 2012. 16, 18, 22, 24, 61

42. J. Jonas and J. Harper, "Effective counterterrorism and the limited role of predictive data mining," Policy Analysis, Tech. Rep. 584, 2006. 16

43. J. Efremova1, B. Ranjbar-Sahraei, F. A. Oliehoek, T. Calders, and K. Tuyls, "A baseline method for genealogical entity resolution," in *Proceedings of the Workshop on Population Reconstruction*, Amsterdam, Netherlands, 2014. 16

44. Z. Fu, M. Boot, P. Christen, and J. Zhou, "Automatic record linkage of individuals and households in historical census data," *accepted by International Journal of Humanities and Arts Computing*, 2014. 16, 18, 29, 75

45. G. Weber, S. Murphy, A. McMurry, D. Macfadden, D. Nigrin, S. Churchill, and I. Kohane, "The Shared Health Research Information Network (SHRINE): a prototype federated query tool for clinical data repositories," *Journal of the American Medical Informatics Association*, vol. 16, no. 5, pp. 614–630, 2009. 16

46. P. Christen, "Febrl: an open source data cleaning, deduplication and record linkage system with a graphical user interface," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Las Vegas, 2008, pp. 1065–1068. 16, 19, 41, 61

47. ——, "Development and user experiences of an open source data cleaning, deduplication and record linkage system," *ACM SIGKDD Explorations*, vol. 11, no. 1, pp. 39–48, 2009. 16, 19, 32

48. P. Christen and D. Belacic, "Automated probabilistic address standardisation and verification," in *Proceedings of Australasian Data Mining Conference*, 2005, pp. 53–68. 16, 19

49. P. Jurczyk, J. Lu, L. Xiong, J. Cragan, and A. Correa, "Fril: A tool for comparative record linkage," in *Proceedings of the AMIA Annual Symposium*, 2008, pp. 440–444. 16, 19

50. H. Kang, L. Getoor, B. Shneiderman, M. Bilgic, and L. Licamele, "Interactive entity resolution in relational data: a visual analytic tool and its evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 5, pp. 999–1014, 2008. 16

51. C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006. 16, 27, 72, 99

52. X. Wu, V. Kumar, J. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z. H. Zhou, M. Steinbach, D. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008. 16

53. W. Fan, X. Jia, J. Li, and S. Ma, "Reasoning about record matching rules," in *Proceedings of the VLDB Endowment*, vol. 2, no. 1, 2009, pp. 407–418. 16

54. Y. Jiang, C. Lin, W. Meng, C. Yu, A. M. Cohen, and N. R. Smalheiser, "Rule-based deduplication of article records from bibliographic databases," *Database*, vol. article no: bat086, 2014. 16

55. Z. Fu, P. Christen, and M. Boot, "Automatic cleaning and linking of historical census data using household information," in *Proceedings of the 15th International Workshop on Domain Driven Data Mining*, Vancouver, Canada, 2011, pp. 413–420. 16, 66, 104, 105

56. E. Borges, M. Carvalhob, R. Galantea, and M. Goncalvesb, "An unsupervised heuristic-based approach for bibliographic metadata deduplication," *Information Processing and Management*, vol. 47, no. 5, pp. 706–718, 2011. 16

57. M. Bilenko, S. Basu, and M. Sahami, "Adaptive product normalization: Using online learning for record linkage in comparison shopping," in *Proceedings of the 5th International Conference on Data Mining*, 2006, pp. 58–65. 16

58. S. Sarawagi and A. Bhamidipaty, "Interactive deduplication using active learning," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002. 16

59. W. E. Winkler, "Overview of research linkage and current research directions," US Bureau of the Census, Statistical Research Report Series RRS2006/02, 2006. 16, 23, 32

60. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007. 16, 23

61. P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," *IEEE Transactions on Knowledge and Data Engineering*, 2012. 16, 23, 56

62. C. Bartini and M. Scannapieco, *Data quality: concepts, methodologies and techniques (Data-Centric Systems and Applications)*. Springer, 2006. 18

63. J. Han and M. Kamber, *Data mining: Concepts and Techniques.*, 2nd ed. Morgan Kaufmann, 2006. 18, 27

64. T. N. Herzog, F. Scheuren, and W. E. Winkler, *Data quality and record linkage techniques*. Springer, 2007. 18

65. G. Bloothooft and M. Schraagen, "Learning name variants from true person resolution," in *Proceedings of the Workshop on Population Reconstruction*, Amsterdam, Netherlands, 2014. 18

66. P. Christen, T. Churches, and J. X. Zhu, "Probabilistic name and address cleaning and standardisation," in *Proceedings of the Australasian Data Mining Workshop*, 2002, pp. 130–145. 19

67. T. Churches, P. Christen, K. Lim, and J. X. Zhu, "Preparation of name and address data for record linkage using hidden Markov models," *BMC Medical Informatics and Decision Making*, vol. 2, no. 9, 2002. 19

68. F. Morillo, J. Aparicio, and L. M. Borja Gonzalez-Albo, "Towards the automation of address identification," *Scientometrics*, vol. 94, no. 1, pp. 207–224, 2013. 19

69. H. Guo, H. Zhu, Z. Guo, X. Zhang, and Z. Su, "Address standardization with latent semantic association," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 1155–1164. 19

70. M. A. Jaro, "Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida," *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 414–420, 1989. 20, 24

71. M. A. Hernandez and S. J. Stolfo, "Real-world data is dirty: Data cleansing and the merge/purge problem," *Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 9–37, 1998. 20, 21

72. R. Baxter, P. Christen, and T. Churches, "A comparison of fast blocking methods for record linkage," in *ACM SIGKDD'03 Workshop on Data Cleaning, Record Linkage and Object Consolidation*, 2003, pp. 25–27. 20, 21

73. A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 169–178. 20, 21, 22

74. A. Aizawa and K. Oyama, "A fast linkage detection scheme for multi-source information integration," in *Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration*, 2005, pp. 30–39. 20, 22

75. P. Lehti and P. Frankhauser, "A precise blocking method for record linkage," in *Proceedings of the International conference on data warehousing and knowledge discovery*, 2005, pp. 210–220. 21

76. T. D. Vries, H. Ke, S. Chawla, and P. Christen, "Robust record linkage blocking using suffix arrays," in *Proceeding of the 18th ACM Conference on Information and Knowledge Management*, 2009, pp. 305–314. 22

77. M. Michelson and C. A. Knoblock, "Learning blocking schemes for record linkage," in *Proceedings of the 21st national conference on Artificial intelligence*, 2006, pp. 440–445. 22

78. S. E. Whang, D. Menestrina, G. Koutrika, M. Theobald, and H. Garcia-Molina, "Entity resolution with iterative blocking," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, 2009, pp. 219–232. 22

79. L. Philips, "The double metaphone search algorithm," *C/C++ Users Journal*, vol. 18, no. 6, pp. 38–43, 2000. 23

80. P. Christen, "A comparison of personal name matching: Techniques and practical issues," in *Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, Hong Kong, 2006, pp. 290–294. 23, 25

81. J.R.Ullmann, "A binary n-gram technique for automatic correction of substitution,deletion, insertion, and reversal errors in words," *Computer Journal*, vol. 20, no. 2, pp. 141–147, 1977. 25

82. L. Gravano, P. G. Ipeirotis, H. V. Jagadish, N. Koudas, S. Muthukrishnan, L. Pietarinen, and D. Srivastava, "Using q-grams in a dbms for approximate string processing," *IEEE Data Engineering Bulletin*, vol. 24, no. 4, pp. 28–34, 2001. 25

83. E. H. Porter and W. E. Winkler, "Approximate string comparison and its effect on an advanced record linkage system," U.S. Bureau of the Census, Tech. Rep., 1997. 26

84. S.B.Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, pp. 249–268, 2007. 26

85. R. O. Duda and P. E. Hart, *Pattern Classification*. Wiley, 2000. 27

86. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Academic Press, 2009. 27

87. N. C. J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000. 27

88. B. Scholkopf and A. Smola, *Learning with Kernels*. MIT Press, 2002. 27

89. M. Michalowski, S. Thakkar, and C. A. Knoblock, "Automatically utilizing secondary sources to align information across sources," *AI Magazine*, vol. 1, pp. 33–44, 26. 27

90. M. Elfeky, V. Verykios, and A. Elmagarmid, "Tailor: A record linkage toolbox," in *Proceedings of the 18th International Conference on Data Engineering*, 2002, pp. 17–28. 27, 28

91. J. Brown, J. Holmes, K. Shah, K. Hall, R. Lazarus, and R. Platt, "Distributed health data networks: a practical and preferred approach to multi-institutional evaluations of comparative effectiveness, safety, and quality of care," *Medical Care*, vol. 48, no. 6 Suppl, pp. 45–51, 2010. 27

92. D. Wilson, "Beyond probabilistic record linkage: Using neural networks and complex features to improve genealogical record linkage," in *Proceedings of the 2011 International Joint Conference on Neural Networks*, 2011, pp. 9–14. 27

93. B. Pixton and C. Giraud-Carrier, "Using structured neural networks for record linkage," in *Proceedings of the 6th Annual Workshop on Technology for Family History and Genealogical Research*, 2006. 27

94. M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington DC, 2003, pp. 39–48. 27

95. P. Christen, "Automatic training example selection for scalable unsupervised record linkage," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Osaka, Japan, 2008, pp. 511–518. 27

96. V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995. 27, 74, 115

97. U. Y. Nahm, M. Bilenko, and R. J. Mooney, "Two approaches to handling noisy variation in text mining," in *Proceedings of the ICML-2002 Workshop on Text Learning*, 2002, pp. 18–27. 27

98. D. V. Kalashnikov and S. Mehrotra, "Domain-independent data cleaning via analysis of entity-relationship graph," *ACM Transactions on Database Systems*, vol. 31, no. 2, pp. 716–767, 2006. 28, 70

99. R. Nuray-Turan, D. V. Kalashnikov, and S. Mehrotra, "Self-tuning in graph-based reference disambiguation," in *Proceedings of the International Conference on Database Systems for Advanced Applications*, 2007, pp. 325–336. 28

100. S. E. Fienberg and D. Manrique-Vallier, "Integrated methodology for multiple systems estimation and record linkage using a missing data formulation," *AStA Advances in Statistical Analysis*, vol. 93, no. 1, pp. 49–60, 2009. 29

101. J. Asher, S. Fienberg, E. Stuart, and A. Zaslavsky, "Inferences for finite populations using multiple data sources with different reference times," in *Proceedings of Statistics Canada Symposium 2002: Modelling Survey Data For Social and Economic Research*, 2003. 29

102. B.-W. On, N. Koudas, D. Lee, and D. Srivastava, "Group linkage," in *Proceeding of the IEEE International Conference on Data Engineering*, Istanbul, Turkey, 2007, pp. 496–505. 29, 71, 73, 87, 101, 133

103. B.-W. On, E. Elmaciogl, D. Lee, J. Kang, and J. Pei, "Improving grouped-entity resolution using quasi-cliques," in *Proceedings of the IEEE International Conference on Data Mining*, Hong Kong, 2006, pp. 1008–1015. 29

104. R. Hall and S. Fienberg, "Valid statistical inference on automatically matched files," in *Proceedings of the 2012 international conference on Privacy in Statistical Databases*, 2012, pp. 131–142. 30

105. P. Ravikumar and W. W. Cohen, "A hierarchical graphical model for record linkage," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, 2004, pp. 454–461. 30

106. T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol. 89, pp. 31–71, 1997. 30, 112

107. S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, Vancouver, Canada, 2003, pp. 561–568. 30

108. Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *Journal of Machine Learning Research*, vol. 5, 2004. 30, 139

109. Y. Xiao, B. Liu, L. Cao, J. Yin, and X. Wu, "SMILE: A similarity-based approach for multiple instance learning," in *Proceedings of the IEEE International Conference on Data Mining*, Sydney, 2010, pp. 309–313. 30, 139

110. Y. Chen, J. Bi, and J. Wang, "MILES: Multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1931–1947, 2006. 30, 112, 113, 114, 139

111. Z. Fu, A. Robles-Kelly, and J. Zhou, "MILIS: Multiple instance learning with instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 958–977, 2011. 30, 112, 113, 114, 120, 123, 127, 139

112. K. Schürer and M. Woollard, *The national sample of the 1881 census of Great Britain. A user Guide and workbook*. Colchester, 2000. 31

113. K. Schürer and W. Woolard, "National sample from the 1881 census of great britain 5% random sample," University of Essex, Historical Censuses and Social Survey Research Group, 2002. 31

114. M. D. Larsen and D. B. Rubin, "Iterative automated record linkage using mixture models," *American Statistical Association*, vol. 79, pp. 32–41, 2001. 32, 33

115. R. Vick and L. Huynh, "The effects of standardizing names for record linkage: Evidence from the United States and Norway," *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, vol. 44, no. 1, pp. 15–24, 2011. 32, 33

116. R. Goeken, L. Huynh, T. A. Lynch, and R. Vick, "New methods of census record linking," *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, vol. 44, no. 1, pp. 7–14, 2011. 33

117. L. Antonie, K. Inwood, D. Lizotte, and J. Ross, "Tracking people over time in 19th century canada," *Machine Learning*, vol. 95, no. 1, pp. 129–146, 2014. 33

118. L. Antonie, K. Inwood, and J. A. Ross, "Dancing with dirty data: Problems in the extraction of life-course evidence from historical censuses," in *Proceedings of the Workshop on Population Reconstruction*, Amsterdam, Netherlands, 2014. 33

119. J. Jefferies, "The UK population: past, present and future," *Focus on People and Migration*, 2005. 35

120. M. Woollard, "The classification of occupations in the 1881 census of England and Wales," Historical Censuses and Social Surveys Research Group, University of Essex, Technical Report, 1999. 50

121. G. Kirby, J. Carson, F. Dunlop, C. Dibben, A. Dearle, L. Williamson, E. Garrett, and A. Reid, "Automatic methods for coding historical occupation descriptions to standard classifications," in *Proceedings of the Workshop on Population Reconstruction*, Amsterdam, Netherlands, 2014. 50

122. P. Christen, *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, 2012. 56

123. W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *Proceedings of the IJCAI-03 Workshop on Information Integration*, 2003, pp. 73–78. 61

124. M. Herschel and F. Naumann, "Scaling up duplicate detection in graph data," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, Napa Valley, California, 2008, pp. 1325–1326. 70

125. P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson Addison-Wesley, 2005. 72

126. G. Chartrand, *Introductory Graph Theory*. Dover Publications, 1985. 72

127. C.-C. Chang and C.-J. Lin, "Libsvm : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, p. 27, 2011. 78

128. T. Caelli and T. Caetano, "Graphical models for graph matching: Approximate models and optimal algorithms," *Pattern Recognition Letters*, vol. 26, no. 3, pp. 339–346, 2005. 96

129. L. Zager and G. Verghese, "Graph similarity scoring and matching," *Applied Mathematics Letters*, vol. 21, no. 1, pp. 86–94, 2008. 97

130. T. Caetano, J. McAuley, L. Cheng, Q. V. Le, and A. Smola, "Learning graph matching," *IEEE TPAMI*, vol. 31, no. 6, pp. 1048–1058, 2009. 97, 99

131. D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Wiley, 2013. 98

132. J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957. 99

133. J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 281–305, 2012. 101

134. Z. Fu, J. Zhou, P. Christen, and M. Boot, "Multiple instance learning for group record linkage," in *Proceedings of the 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Kuala Lumpur, Malaysia, 2012, pp. 171–182. 108

135. F. Li and C. Sminchisescu, "Convex multiple instance learning by estimating likelihood ratio," in *Proceedings of the Conference on Advances in Neural Information Processing Systems*, 2010. 112

136. R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008. 123