A Bayesian Framework for Duplicate Detection, Record Linkage, and Subsequent Inference with Linked Files

Mauricio Sadinle

Carnegie Mellon University

October 3, 2013

Abstract

Record linkage techniques allow us to combine different sources of information from a common population in the absence of unique identifiers. Linking multiple files is an important task in a wide variety of applications, since it permits to gather information that would not be otherwise available, or that would be too expensive to collect. In practice, an additional complication appears when the datafiles to be linked contain duplicates. The final output of duplicate detection and record linkage techniques is traditionally a single file on which different analyses are carried out. Performing any analysis on those linked files completely ignores the uncertainty associated with the record linkage and duplicate detection decisions. The reason for this practice is that traditionally record linkage and duplicate detection techniques do not provide a proper account of the uncertainty of their outputs.

We present a class of inferential scenarios where uncertainty from duplicate detection/record linkage decisions can be incorporated into subsequent statistical analyses, in such a way that both steps can be done sequentially. By taking this two-step approach we can solve the record linkage/duplicate detection step using comparison data, which avoids the explicit modeling of fields that are only important for record linkage and duplicate detection, such as names and addresses. This approach however will not lead to valid inferences in most scenarios, and therefore we explore conditions under which the inference can be done sequentially. For this approach to work, we propose methods for finding duplicates in a datafile, and for linking multiples files potentially containing duplicates, that provide a proper account of the uncertainty of the decisions that they output. The uncertainty from record linkage and duplicate detection is handled as a posterior distribution that can be incorporated in a variety of subsequent analyses, such as the estimation of population sizes or the estimation of relationships between variables that appear in different files.

1 Introduction

Combining different sources of information from the same population is important in a wide variety of applications, including merging post–enumeration surveys and census data for census coverage evaluation (e.g., Winkler, 1988; Jaro, 1989; Winkler and Thibaudeau, 1991), and linking health–care databases for epidemiological studies (e.g., Bell et al., 1994; Méray et al., 2007). It is also common to find duplicates within the datafiles that we want to analyze, and therefore we need to account for the existence of duplicates to avoid biased results.

The task of finding sets of records that refer to the same entity is not trivial when unique identifiers are not available, and specially when records are subject to errors. In this document two or more records referring to the same entity are called *coreferent*. When our aim is to find coreferent records within one datafile, we refer to the task as *duplicate detection*, and when our goal is to find coreferent records across different datafiles, we call the task *record linkage*, although some authors use these and other terms indistinguishably (e.g. see Elmagarmid et al., 2007; Christen, 2012a).

Although, in principle, when linking multiple datafiles we could put all files together in a concatenated file, and treat the problem as one of duplicate detection, we would like to distinguish both tasks in the following sense. When linking multiple files it is important to acknowledge the data collection process that lead to creation of each of the files. For example, the data sources to be linked may come from surveys, administrative registries, convenience samples, or even censuses. Also, the different data collection processes may lead to different kinds of errors in the different files. On the other hand, when finding duplicates within one file, we assume all records have been subject to the same data collection process, and therefore all records may be subject to the same distribution of error.

Current approaches to duplicate detection and record linkage train classifiers or mixture models using comparison data computed on pairs of records, and output independent decisions on the coreference status of each record pair (Elmagarmid et al., 2007; Herzog et al., 2007). This practice does not guarantee transitivity of the coreference decisions and thus require resolving discrepancies in a post–processing step. For example, it may be possible that records i and j are declared as being coreferent, as well as records j and k, but records i and k may be declared as non–coreferent. If i, j, and k truly correspond to the same entity, the non–transitivity could occur due to measurement error or incomplete record information. It may be the case however that only two of those records are coreferent, but current methodologies do not offer any representation of uncertainty in these situations.

Many studies require linking files or detecting duplicates within a file, or both, as a step that precedes some statistical analysis. For example the usual methodology of census coverage evaluation matches a coverage measurement survey to the census data in order to estimate population sizes using dual–system estimation (e.g. Hogan, 1992, 1993). The different types of analysis that follow the linkage step, however, do not usually account for the uncertainty coming from record linkage.

In this thesis we propose new methods that guarantee transitivity of the coreference decisions in the context of finding duplicates within one file, as well as in the context of linking multiple files that may contain duplicates. Our methods provide posterior distributions on the space of coreference decisions, and therefore we can incorporate this uncertainty in subsequent statistical analysis using a Bayesian framework.

The remaining document is organized as follows. Section 2 provides an overview of the proposed framework. Section 3 presents a literature review of current methods for record linkage and duplicate detection, later develops a methodology for duplicate detection, and ends with a proposed generalization for linking multiple datafiles. Section 4 outlines how to incorporate the uncertainty obtained from record linkage and duplicate detection into some statistical procedures. Finally, Section 5 presents the proposed future steps of this thesis, including applications to Human Rights and the US Census.

2 The Proposed Framework

Assume we have K datafiles possibly containing sets of coreferent records within them, as well as across them. Let us denote the kth datafile as $\mathbf{X}_k$, and the number of records that it contains as r_k, $k = 1, \ldots, K$. Each datafile can be conceptualized as a set of records, and therefore we can define $\mathbf{X} = \bigcup_{k=1}^{K} \mathbf{X}_k$ as the concatenated file containing all the records coming from the K different sources. The number of records in the combined file $\mathbf{X}$ is denoted $r = \sum_k r_k$. We label the records in the combined file as $\{1, \ldots, r\}$, such that if file $\mathbf{X}_k$ contains record i, file $\mathbf{X}_{k'}$ contains record i', and $k < k'$, then $i < i'$. In practical terms, this means that file $\mathbf{X}$ organizes the K different files in a common format, and the files are concatenated one under another.

Assuming that there are $n \leq r$ different entities represented in the combined datafile $\mathbf{X}$, we can safely think that $\mathbf{X}$ can be partitioned into n groups of records, where each group represents a set of coreferent records. This partition of the combined file is our parameter of interest in joint duplicate detection and record linkage, and it can be represented by a matrix, as presented in the next section.

2.1 The Coreference Matrix

Let us consider the (unobserved) *coreference matrix* $\mathbf{\Delta}$ of size $r \times r$, whose (i, j)th entry is defined as

$$\Delta_{ij} = \begin{cases} 1, & \text{if records } i \text{ and } j \text{ refer to the same entity;} \\ 0, & \text{otherwise.} \end{cases}$$

This definition implies that $\mathbf{\Delta}$ is symmetric with diagonal entries containing only ones. It also implies that $\mathbf{\Delta}$ can be rearranged as a block–diagonal matrix by permuting its rows and columns, which is equivalent to relabeling the entities in the combined datafile. In the block–diagonal version of $\mathbf{\Delta}$ each block represents a group of coreferent records, and so the total number of blocks equals n, the number of entities represented in $\mathbf{X}$. An important fact that we will use in subsequent sections is that $\text{rank}(\mathbf{\Delta}) = n$.

From a computational point of view, representing partitions using matrices might be inefficient, specially when the combined file $\mathbf{X}$ is large. In practice, we represent a partition by an arbitrary labeling of its elements, but we choose to use coreference matrices in this document to simplify notation and the exposition of ideas.

The labeling of the records in the combined file allows us to express the coreference matrix as a block–matrix as follows:

$$\mathbf{\Delta} = \begin{pmatrix} \mathbf{\Delta}_{11} & \mathbf{\Delta}_{12} & \cdots & \mathbf{\Delta}_{1K} \\ \mathbf{\Delta}_{21} & \mathbf{\Delta}_{22} & \cdots & \mathbf{\Delta}_{2K} \\ \vdots & \vdots & & \vdots \\ \mathbf{\Delta}_{K1} & \mathbf{\Delta}_{K2} & \cdots & \mathbf{\Delta}_{KK} \end{pmatrix},$$

where $\mathbf{\Delta}_{kk}$ is a submatrix of size $r_k \times r_k$ that contains the information of duplicates for file k, and $\mathbf{\Delta}_{kl}$ is a submatrix of size $r_k \times r_l$ that contains the information of links between files k and l. Notice that making inference on the submatrices $\mathbf{\Delta}_{kk}$ represents the task of duplicate detection within files, and similarly, inference on the set of matrices $\mathbf{\Delta}_{kl}$, $k \neq l$, represents the task of linking pairs of files. In subsequent sections we will be using the facts that $\text{rank}(\mathbf{\Delta}_{kk}) = n_k$, where n_k is the number of entities represented in file k, and $\text{rank}(\mathbf{\Delta}_{kl}) = n_{kl}$, where n_{kl} is the number of entities represented in both files k and l.

2.2 Bayesian Inference on the Coreference Matrix

The goal of joint record linkage and duplicate detection is to make inference on the partition of the combined file $\mathbf{X}$, such that coreferent records get grouped in the same element of the partition. Given the lack of unique identifiers, we use the information contained in the file $\mathbf{X}$ for this purpose. From a Bayesian standpoint, we are interested in finding a posterior distribution on the set of partitions of the combined file, represented by coreference matrices, given the information contained in $\mathbf{X}$, this is

$$p(\mathbf{\Delta}|\mathbf{X}) \propto p(\mathbf{\Delta})p(\mathbf{X}|\mathbf{\Delta}), \tag{1}$$

simply by Bayes theorem. Section 3 presents methods for making inference on the coreference matrix under the assumption that pairwise comparisons between records are all the information that we need in order to determine their coreference status, this is

$$p(\mathbf{\Delta}|\mathbf{X}) = p(\mathbf{\Delta}|\mathbf{\Gamma}(\mathbf{X})) \propto p(\mathbf{\Delta})p(\mathbf{\Gamma}(\mathbf{X})|\mathbf{\Delta}), \tag{2}$$

where $\mathbf{\Gamma}(\mathbf{X})$ represents an array of comparisons among pairs of records. In other words, we are assuming $\mathbf{\Gamma}(\mathbf{X})$ to be Bayesian sufficient for $\mathbf{\Delta}$. This assumption is implicit in most of the literature on record linkage and duplicate detection (e.g. Fellegi and Sunter, 1969; Winkler, 1988; Jaro, 1989; Winkler and Thibaudeau, 1991; Larsen and Rubin, 2001; Herzog et al., 2007).

The main advantage of working with the assumption presented in Equation (2), compared to the direct use of the Bayes theorem as in Equation (1), is that models for $\mathbf{\Gamma}(\mathbf{X})|\mathbf{\Delta}$ will often be much simpler than models for $\mathbf{X}|\mathbf{\Delta}$. In effect, depending on the context, the file $\mathbf{X}$ will contain some combination of fields like family and given names, dates, addresses, phone numbers, etc. Modeling $\mathbf{X}|\mathbf{\Delta}$ implies proposing a model for

such fields, which typically involves proposing a model of how such information gets corrupted. Tancredi and Liseo (2011) and more recently Steorts et al. (2013) have addressed the record linkage problem by modeling $\mathbf{X}|\boldsymbol{\Delta}$ directly, but they confine themselves to work only with categorical information, for which there exist models for measurement error, such as the *hit–miss* model of Copas and Hilton (1990). On the other hand, modeling comparison data $\boldsymbol{\Gamma}(\mathbf{X})|\boldsymbol{\Delta}$ can be done in a simple way, as presented in Section 3, as long as the records can be compared in a meaningful way.

2.3 Comparison Data

As the name suggests, comparison data are obtained by comparing pairs of records, with the goal of finding evidence of whether two records refer to the same entity or not. Intuitively, two records referring to the same entity should be very similar. The way to construct the comparisons depends on the information contained by the records. The most straightforward way of comparing the same field of two records is by checking whether their information agree or not. Although this comparison method is extensively used, and it is appropriate for comparing unordered categorical fields (e.g. sex or race), it completely ignores partial agreement among the information being compared.

Winkler (1990) proposes to take into account partial agreement among fields that contain strings (e.g. given names) by computing a string metric, such as the normalized Levenshtein edit distance or any other (see Bilenko et al., 2003; Elmagarmid et al., 2007), and then dividing the resulting set of similarity measures into different levels of agreement. Winkler's approach can be extended to compute levels of agreement for fields that are not appropriately compared in a dichotomous fashion.

We compare the field f of records i and j by computing some similarity measure $\mathcal{S}_f(i,j)$. The range of this similarity measure is then divided into L_f intervals $I_{f1}, I_{f2}, \ldots, I_{fL_f}$, that represent different levels of agreement. By convention, the first interval, I_{f1}, represents the highest level of agreement, which includes total agreement, and the last interval, I_{fL_f}, represents the lowest level of agreement, which depending on the field represents complete or strong disagreement. Based on these intervals, the comparison data consist of ordinal variables that represent levels of agreement. For records i and j, and comparison criterion f, we define

$$\gamma_{ij}^f = l, \text{ if } \mathcal{S}_f(i,j) \in I_{fl}. \tag{3}$$

These different field comparisons are collected in vectors for each pair of records, as in Fellegi and Sunter (1969), although in this case they are not limited to binary comparisons. $\boldsymbol{\gamma}_{ij} = (\gamma_{ij}^1, \ldots, \gamma_{ij}^f, \ldots, \gamma_{ij}^F)$ denotes the comparison vector for records i and j, where F is the number of fields being compared. In Section 3, we assume that the array of comparisons $\boldsymbol{\gamma} = \{\boldsymbol{\gamma}_{ij}\}_{i,j}$ is a realization of a random array $\boldsymbol{\Gamma}(\mathbf{X})$ that depends on the variables in the datafiles. Henceforth we simply denote $\boldsymbol{\Gamma}(\mathbf{X})$ as $\boldsymbol{\Gamma}$.

2.4 Bayesian Inference with Uncertainty on the Coreference Matrix

Record linkage and duplicate detection often precede other inferential procedures, such as population size estimation or regression. In general, we might be interested in estimating a parameter vector $\boldsymbol{\phi}$ using data from the combined file $\mathbf{X}$. From a Bayesian standpoint this requires finding a posterior distribution $p(\boldsymbol{\phi}|\mathbf{X})$. The direct use of the data $\mathbf{X}$ in traditional inferential procedures is not appropriate, since the existence of coreferent records within the file may lead to biased inferences. Inferential procedures therefore need to take into account the uncertainty on the coreference matrix represented by some posterior $p(\boldsymbol{\Delta}|\mathbf{X})$.

To address this inferential challenge, we first consider the simplest case scenario. Let us suppose that we know the true $\boldsymbol{\Delta}$, this is, say we know which records are coreferent. How can we estimate $\boldsymbol{\phi}$ using $\mathbf{X}$?, this is, how can we find $p(\boldsymbol{\phi}|\mathbf{X}, \boldsymbol{\Delta})$? The answer, as usual, depends on the problem, although in general a set of coreferent records can be regarded as multiple measurements on the same entity, and therefore measurement error models (e.g. Fuller, 1987) can play an important role in modeling $\mathbf{X}|\boldsymbol{\phi}, \boldsymbol{\Delta}$, specially if $\mathbf{X}$ contains continuous measurements. For most applications, however, $\mathbf{X}$ might contain categorical variables measured with error, and thus extensions of the *hit–miss* model (Copas and Hilton, 1990) will play an important role. Section 4 presents ideas on how to obtain posteriors $p(\boldsymbol{\phi}|\mathbf{X}, \boldsymbol{\Delta})$ for different types of problems, but for the sake of outlying the general framework, we will assume that we are able to obtain $p(\boldsymbol{\phi}|\mathbf{X}, \boldsymbol{\Delta})$ in a meaningful way.

Let us assume that we have a model M_A for $\mathbf{X}|\boldsymbol{\phi}, \boldsymbol{\Delta}$ that allows us to find $p_A(\boldsymbol{\phi}|\mathbf{X}, \boldsymbol{\Delta})$ using the prior $p(\boldsymbol{\phi})$. In the last paragraph we were conditioning on a known coreference matrix $\boldsymbol{\Delta}$, but in general the uncertainty on the configuration of this matrix may be summarized by a posterior distribution. Therefore,

suppose we have a posterior $p_B(\boldsymbol{\Delta}|\mathbf{X})$ coming from a model M_B, using the prior $p(\boldsymbol{\Delta})$. Naively, we would like to compute

$$p(\boldsymbol{\phi}|\mathbf{X}) = \sum_{\boldsymbol{\Delta}} p_A(\boldsymbol{\phi}|\mathbf{X}, \boldsymbol{\Delta}) p_B(\boldsymbol{\Delta}|\mathbf{X}) \tag{4}$$

in order to account for uncertainty on the coreference matrix. However, models M_A and M_B may not be compatible. Notice that model M_A, along with priors $p(\boldsymbol{\Delta})$ and $p(\boldsymbol{\phi})$ necessarily imply

$$p_A(\boldsymbol{\Delta}|\mathbf{X}) \propto p(\boldsymbol{\Delta}) \int_{\boldsymbol{\phi}} p_A(\mathbf{X}|\boldsymbol{\phi}, \boldsymbol{\Delta}) p(\boldsymbol{\phi}) d\boldsymbol{\phi}$$

which in general will not be the same as $p_B(\boldsymbol{\Delta}|\mathbf{X})$. Since models M_A and M_B do not necessarily fit together in a common joint distribution of $\mathbf{X}$, $\boldsymbol{\phi}$, and $\boldsymbol{\Delta}$, Equation (4) will in general lead to non valid inferences. Notice that this issue also occurs in multiple imputation (Rubin, 1987) whenever the model used to impute missing data is different from the model used to analyze the multiple imputed datasets.

A natural solution to the above difficulty is to model $\mathbf{X}|\boldsymbol{\phi}, \boldsymbol{\Delta}$ and then make inference on $\boldsymbol{\phi}$ and $\boldsymbol{\Delta}$ simultaneously. The advantage of this approach is that it is internally consistent and it will lead to valid posteriors $p(\boldsymbol{\phi}|\mathbf{X})$. Such approach is taken by Tancredi and Liseo (2011) in order to simultaneously link datafiles and estimate population sizes using capture–recapture methods. As we mentioned in Section 2.2, however, it might be difficult to model $\mathbf{X}$, since it usually contains names, addresses, phone numbers, dates, etc, and therefore authors taking this approach have focused only on categorical information. Furthermore, the vector of interest $\boldsymbol{\phi}$ may involve directly none or only some of the variables in $\mathbf{X}$, and therefore the complete modeling of $\mathbf{X}$ might be unnecessary.

In this document we propose to explore inferential scenarios where the application of the simple formula given by Equation (4) leads to valid inferences, where $p_A(\phi|\boldsymbol{\Delta}, X)$ and $p_B(\boldsymbol{\Delta}|X)$ come from different models. For instance, the posterior on the space of partitions (coreference matrices) can be obtained using comparison data, as it will be presented in Section 3, and for a given partition the posterior on the parameters of interest can be obtained using existing models or perhaps some simple modification.

We now present some simple conditions under which Equation (4) leads to valid inferences.

Condition 1. If $\boldsymbol{\phi} \perp \mathbf{X}|\boldsymbol{\Delta}$ then Equation (4) leads to valid inferences. Here Equation (4) simplifies to

$$p(\boldsymbol{\phi}|\mathbf{X}) = \sum_{\boldsymbol{\Delta}} p_A(\boldsymbol{\phi}|\boldsymbol{\Delta}) p_B(\boldsymbol{\Delta}|\mathbf{X})$$

Condition 1 deals with cases where inference on $\boldsymbol{\phi}$ only directly involves $\boldsymbol{\Delta}$. Under Condition 1 we will be able to make inference on population sizes using multiple systems estimation/ capture–recapture methods, accounting for uncertainty in record linkage and duplicate detection, as it will be presented in Section 4.

Condition 2. Say $\mathbf{X} = (\mathbf{X}^1, \mathbf{X}^2)$. If $\boldsymbol{\phi} \perp \mathbf{X}^2|\boldsymbol{\Delta}, \mathbf{X}^1$, and $\boldsymbol{\Delta} \perp \mathbf{X}^1|\mathbf{X}^2$ then Equation (4) leads to valid inferences. In this case Equation (4) can be written as

$$p(\boldsymbol{\phi}|\mathbf{X}) = \sum_{\boldsymbol{\Delta}} p_A(\boldsymbol{\phi}|\boldsymbol{\Delta}, \mathbf{X}^1) p_B(\boldsymbol{\Delta}|\mathbf{X}^2)$$

An intuitive way to interpret Condition 2 is that once the set of variables $\mathbf{X}^2$ is being used to estimate $\boldsymbol{\Delta}$, the set $\mathbf{X}^1$ does not provide further information, and the parameter vector of interest $\boldsymbol{\phi}$ involves only $\mathbf{X}^1$. An extreme example of $\mathbf{X}^2$ would be if it contained a unique identifier measured without error, since this is all we need to determine $\boldsymbol{\Delta}$. In such a situation, of course, the record linkage and duplicate detection processes would be trivial.

Lahiri and Larsen (2005) consider the regression problem where response and covariates appear in different files, but both files share a set of fields that are useful to link them. In this scenario, response and covariates are not useful for record linkage, but the regression involving them should account for uncertainty in the linkages. This scenario fits under Condition 2, and we discuss it further in Section 4.

3 Duplicate Detection and Record Linkage

3.1 Literature Review

Duplicate detection and record linkage techniques have the goal of finding coreferent records in two traditional scenarios. The first one involves one single datafile that contains duplicates, and the goal is to detect them.

The second scenario involves two files that have an overlap in terms of the entities that they represent. The goal in this case is to identify the records referring to entities represented in both files. The last scenario traditionally assumes that the files contain no duplicates.

In the context of duplicate detection, there are $r(r-1)/2$ pairs that need to be classified into coreferent and non–coreferent pairs, where r is the size of the file. When linking two files, the number of pairs is $r_1 \times r_2$, where r_1 is the size of the first file, and r_2 the size of the second one. In either case, pairwise comparison data are the input of the method.

The existing methods for probabilistic duplicate detection and record linkage can be roughly classified in two groups, those unsupervised that use mixture models, and those supervised that use classification methods. A thorough survey on methods for duplicate detection and record linkage is presented by Elmagarmid et al. (2007). We now present the general idea of those two groups of methods.

3.1.1 Finding Coreferent Record Pairs Using Mixture Models

In this approach we assume that the comparison vector $\boldsymbol{\gamma}_{ij}$ is a realization of a random vector $\boldsymbol{\Gamma}_{ij}$, and the comparison data $\boldsymbol{\gamma} = \{\boldsymbol{\gamma}_{ij}\}_{i,j}$ are a realization of a random array $\boldsymbol{\Gamma}$. Since we expect coreferent records to largely agree in the information that they contain, we assume that the distribution of $\boldsymbol{\Gamma}_{ij}$ is the same for all record pairs that refer to the *same* entity (regardless the entity), and that the distribution of $\boldsymbol{\Gamma}_{ij}$ is the same for all record pairs that refer to *different* entities (regardless the pair of entities).

The above intuitive description can be formalized into a model for the comparison data as

$$\boldsymbol{\Gamma}_{ij}|\Delta_{ij} = 1 \overset{iid}{\sim} G_1, \qquad \boldsymbol{\Gamma}_{ij}|\Delta_{ij} = 0 \overset{iid}{\sim} G_0, \tag{5}$$

where G_1 and G_0 represent the models of the comparison data for pairs that are coreferent and not coreferent, respectively. These models may change depending on the comparison data at hand, this is, depending on the availability of binary comparisons, similarity measures, etc.

The key component of the mixture model implementation is that in addition to the model of Equation (5), the Δ_{ij}'s are modeled as i.i.d. Bernoulli(p). The Δ_{ij}'s, along with the parameters of the complete model are usually estimated using the EM algorithm. If the distribution of the $\boldsymbol{\Gamma}_{ij}$'s among coreferent records is well separated from the distribution of the $\boldsymbol{\Gamma}_{ij}$'s among non–coreferent records, the separation of the record pairs into coreferent and non–coreferent will be of good quality.

This approach for finding coreferent records was initially proposed by Fellegi and Sunter (1969) in the record linkage context, and its modern implementations using the EM algorithm have been widely used (e.g. Winkler, 1988; Jaro, 1989; Larsen and Rubin, 2001). Fellegi and Sunter (1969) proposed a decision rule that allows us to separate pairs of records into matches, non–matches, and possible matches that are sent for clerical review. The rule of Fellegi and Sunter is optimal in the sense that it minimizes the probability of assigning a pair to the subset of possible matches, subject to two user–defined tolerable levels of error: the probability of false matches, and the probability of false non–matches.

This approach outputs independent decisions on the coreference status of pairs of records, and it relies on the Δ_{ij}'s being independent, which is obviously violated in this context. The result of this methodology therefore may not satisfy transitivity or other coherent constraints in the problem. For example, in the case of linking files, if we assume that the files do not contain duplicates, then each record in the first file may be linked to maximum one other record in the second file, and viceversa. When solving the record linkage problem using this approach, there is nothing that enforces this maximum–one–to–one requirement in the model itself, and therefore some post processing steps are required (see Jaro, 1989). Similarly, in the case of finding duplicates, we may obtain non–transitive coreference decisions that have to be reconciled somehow. More recently, Sadinle and Fienberg (2013) extended this approach to the linkage of more that two datafiles under the assumption that the files do not contain duplicates, which is quite restrictive in practice. The approach of Sadinle and Fienberg (2013) naturally inherits the before mentioned difficulties of the mixture model implementation. In Section 3.12 we propose a solution to the problem of linking multiple datafiles that may contain duplicates. See also Steorts et al. (2013) for an alternative approach.

3.1.2 Finding Coreferent Record Pairs Using Classification Methods

The problem of finding coreferent pairs is a classification problem: we need to separate record pairs into coreferent and non–coreferent classes. If we have access to a sample of record pairs for which the true coreference status is known, we can train a classifier on this sample, and then predict the coreference status

of the remaining record pairs (e.g. Cochinwala et al., 2001; Bilenko et al., 2003; Christen, 2008; Ventura et al., 2013).

Classification methods typically assume that we are dealing with i.i.d. data, and therefore the training of the models and the prediction using them rely heavily on this assumption. The fact that these methods output independent coreference decisions for pairs of records imply that this approach may lead to conflicting decisions in either duplicate detection or in record linkage contexts, this is, we may obtain pairs of records with non–transitive decisions when finding duplicates, and may violate the maximum–one–to–one assignment constraint of traditional record linkage scenarios. Typically some subsequent post processing step is required to solve these inconveniences.

In the subsequent subsections we will propose some methods that guarantee the transitivity of the coreference decisions. In Sections 3.2 through 3.11 we propose a methodology for finding duplicates within one datafile, and therefore r denotes the number of records that the file contains. Section 3.12 extends the proposed methodology to the context where we want to link multiple files that potentially contain duplicates.

3.2 Duplicate Detection Using a Bayesian Partitioning Model

In this section we revisit the model presented in Equation (5), since we believe that it is a reasonable way to approach the problem of finding coreferent records. We however propose a way to correct the difficulty of the mixture model implementation, this is, we do not assume that the Δ_{ij}'s are i.i.d. Bernoulli(p), but instead we treat the Δ_{ij}'s as the entries of a coreference matrix that in turn represents a partition.

Leaving G_1 and G_0 unspecified in the model of Equation (5), we can see that for a configuration $\boldsymbol{\delta}$ of the coreference matrix $\boldsymbol{\Delta}$, the joint probability of observing the comparison data $\boldsymbol{\gamma}$ can be written as

$$\mathbb{P}(\boldsymbol{\Gamma} = \boldsymbol{\gamma} | \boldsymbol{\Delta} = \boldsymbol{\delta}) \quad = \quad \prod_{i<j} \mathbb{P}_1(\boldsymbol{\gamma}_{ij})^{\delta_{ij}} \mathbb{P}_0(\boldsymbol{\gamma}_{ij})^{1-\delta_{ij}}, \tag{6}$$

where δ_{ij} represents the (i,j)th element of $\boldsymbol{\delta}$. Up to this point, this is the same as in a mixture model implementation, but here $\boldsymbol{\Delta}$ takes values on the space of partitions. Bayesian inference requires to use a prior distribution on the set of possible configurations of the coreference matrix $\boldsymbol{\Delta}$, this is, a prior on the space of partitions of the file.

3.3 Prior Distribution of the Coreference Matrix

In this section we present the prior distribution of the matrix $\boldsymbol{\Delta}$ used in this document. We denote $\tilde{r}$ as the number of unique records in the datafile, this is, $\tilde{r}$ is the number of records discounting exact duplicates. We assume $\tilde{r}$ is the maximum number of entities possibly represented in the datafile, since we expect not to have further information available to distinguish exact duplicates. By assigning an arbitrary labeling to the $\tilde{r}$ potential entities, we can introduce the vectors $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{iq}, \ldots Z_{i\tilde{r}})^T$, $i = 1, \ldots, r$, where

$$Z_{iq} = \begin{cases} 1, & \text{if record } i \text{ represents entity } q; \\ 0, & \text{otherwise;} \end{cases}$$

and let $\mathbf{Z}$ be the $r \times \tilde{r}$ matrix containing the vectors $\mathbf{Z}_i$ in its rows. Notice that although the ordering of the $\tilde{r}$ potentially existing entities is arbitrary, any permutation of the columns of $\mathbf{Z}$ leads to the same partition of the records. Also notice that if the number of entities n is lower than $\tilde{r}$, then $\mathbf{Z}$ will have $\tilde{r} - n$ columns of zeroes. The usefulness of the preceding representation comes from the fact that if records i and j represent the same entity, then $\mathbf{Z}_i^T \mathbf{Z}_j = 1$, otherwise $\mathbf{Z}_i^T \mathbf{Z}_j = 0$, which implies that $\boldsymbol{\Delta} = \mathbf{Z}\mathbf{Z}^T$. Since $\mathbf{Z}\mathbf{Z}^T$ is invariant to permutations of the columns of $\mathbf{Z}$, we can safely obtain a prior for $\boldsymbol{\Delta}$ by proposing a prior for the $\mathbf{Z}_i$ vectors. Also, since $\text{rank}(\mathbf{Z}) = \text{rank}(\boldsymbol{\Delta}) = n$, a prior for the $\mathbf{Z}_i$ vectors implies a prior for the number of entities represented in the datafile. In this document we propose to model the $\mathbf{Z}_i$'s a priori as a random sample from a multinomial distribution with vector of probabilities $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_{\tilde{r}})$, which represents the entities' relative propensities to appear in the datafile. We can also impose a hyperprior for $\boldsymbol{\pi}$, which is here taken as a symmetric Dirichlet distribution with parameter vector $\alpha \mathbf{1}_{\tilde{r}}$, where $\mathbf{1}_{\tilde{r}}$ is a vector of ones of length $\tilde{r}$, and α is a positive constant. The symmetric Dirichlet distribution treats equally each potential entity since we typically cannot distinguish them a priori. The parameter α controls the concentration of the distribution of $\boldsymbol{\pi}$ around $\tilde{r}^{-1}\mathbf{1}_{\tilde{r}}$. If $\alpha = 1$, then the vector of the entities' propensities to appear in the datafile ($\boldsymbol{\pi}$) is uniformly distributed on the standard $(\tilde{r} - 1)$–simplex; if $\alpha < 1$, as $\alpha \to 0$, the prior of $\boldsymbol{\pi}$ gets more concentrated around the vertices of the simplex, which represents the prior belief that there is one

entity overly represented in the datafile; and finally, if $\alpha > 1$, as $\alpha \to \infty$, the prior of $\boldsymbol{\pi}$ converges to a point mass located at $\tilde{r}^{-1}\mathbf{1}_{\tilde{r}}$.

The preceding $\mathbf{Z}_i$'s prior leads to a distribution on the set of configurations of $\boldsymbol{\Delta}$, where each of these configurations represents a partition of r objects. This distribution is known as the Dirichlet multinomial model for partitions (Keener et al., 1987), or the Dirichlet partition model (McCullagh, 2011). Keener et al. (1987) study several properties of the Dirichlet partition model, and we refer the reader to their article for further details.

3.4 Model Restatement

The model in Equation (5) along with the prior of $\boldsymbol{\Delta}$ can be rewritten in a hierarchical form as

$$\begin{aligned} \boldsymbol{\pi} &\sim \text{Dirichlet}(\alpha\mathbf{1}_{\tilde{r}}), & \mathbf{Z}_i|\boldsymbol{\pi} &\overset{iid}{\sim} \text{Multinomial}(1, \boldsymbol{\pi}), \\ \boldsymbol{\Gamma}_{ij}|\mathbf{Z}_i^T\mathbf{Z}_j = 1 &\overset{iid}{\sim} G_1, & \boldsymbol{\Gamma}_{ij}|\mathbf{Z}_i^T\mathbf{Z}_j = 0 &\overset{iid}{\sim} G_0. \end{aligned} \tag{7}$$

The posterior distribution of $\boldsymbol{\Delta}$ can be obtained from the posterior of $\mathbf{Z}$, which can be approximated using Markov chain Monte Carlo (MCMC) methods.

3.5 A Model for Independent Comparison Fields

In this section we describe a simple parametrization for G_1 and G_0. We present a model for comparisons that use levels of agreement (as in Section 2.3), assuming that the comparison fields are independent for both coreferent and non–coreferent records, or in other terms, $\Gamma_{ij}^f \perp \Gamma_{ij}^{f'}|\Delta_{ij} = 1$ and $\Gamma_{ij}^f \perp \Gamma_{ij}^{f'}|\Delta_{ij} = 0$ for all f and f', $f \neq f'$.

If comparison f is divided into L_f levels of agreement, the distribution of the agreement levels Γ_{ij}^f among coreferent records can be modeled according to a multinomial distribution, this is

$$\mathbb{P}_1(\Gamma_{ij}^f = \gamma_{ij}^f) = \prod_{l=1}^{L_f} (m_{fl}^*)^{I(\gamma_{ij}^f = l)} \tag{8}$$

where γ_{ij}^f represents an observed level of agreement, $m_{fl}^* = \mathbb{P}_1(\Gamma_{ij}^f = l)$, and $\sum_{l=1}^{L_f} m_{fl}^* = 1$. It is easy to show that these probabilities can be rewritten as

$$m_{fl}^* = \begin{cases} m_{f1}, & \text{if } l = 1; \\ m_{fl}\prod_{h<l}(1 - m_{fh}), & \text{if } 1 < l < L_f; \\ \prod_{h<L_f}(1 - m_{fh}), & \text{if } l = L_f; \end{cases} \tag{9}$$

where $m_{f1} = \mathbb{P}_1(\Gamma_{ij}^f = 1)$, and $m_{fl} = \mathbb{P}_1(\Gamma_{ij}^f = l|\Gamma_{ij}^f > l - 1)$ for $1 < l < L_f$. We choose to parameterize the distribution of Γ_{ij}^f among coreferent records in terms of the conditional probabilities m_{fl} since this parametrization facilitates prior specification, as presented in Section 3.7. Using this parametrization, the model in Equation (8) can be reexpressed as

$$\mathbb{P}_1(\boldsymbol{\Gamma}_{ij}^f = \boldsymbol{\gamma}_{ij}^f) = \prod_{l=1}^{L_f - 1} m_{fl}^{I(\gamma_{ij}^f = l)}(1 - m_{fl})^{I(\gamma_{ij}^f > l)}. \tag{10}$$

Notice that if $L_f = 2$, this is, comparison f is binary, we obtain the traditional binary comparisons model. We follow an analogous construction of the distribution of Γ_{ij}^f among non–coreferent pairs, in which case $u_{f1} = \mathbb{P}_0(\Gamma_{ij}^f = 1)$, and $u_{fl} = \mathbb{P}_0(\Gamma_{ij}^f = l|\Gamma_{ij}^f > l - 1)$ for $1 < l < L_f$.

3.6 Missing Comparisons and Conditional Independence

The combination of the assumptions of the comparison fields being conditionally independent (CI), and the comparisons being missing at random (MAR), make it straightforward to deal with missing comparisons. In fact, under these assumptions

$$\mathbb{P}_1(\boldsymbol{\gamma}_{ij}^{obs}|\Phi_1) = \prod_{f=1}^{F} \left[\prod_{l=1}^{L_f - 1} m_{fl}^{I(\gamma_{ij}^f = l)}(1 - m_{fl})^{I(\gamma_{ij}^f > l)} \right]^{I_{obs}(\gamma_{ij}^f)}, \tag{11}$$

where $I_{obs}(\cdot)$ is one if its argument is observed, and zero it is missing, and $\Phi_1 = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_F)$, with $\boldsymbol{m}_f = (m_{f1}, \ldots, m_{f,L_f-1})$. A similar expression is obtained for $\mathbb{P}_0(\boldsymbol{\gamma}_{ij}^{obs}|\Phi_0)$ where $\Phi_0 = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_F)$, with $\boldsymbol{u}_f = (u_{f1}, \ldots, u_{f,L_f-1})$. The above equation indicates that the combination of the CI and MAR assumptions allow us to ignore the comparisons that are not observed, and yet model the observed comparisons in a simple fashion.

Under the CI assumption we can write $\mathbb{P}(\boldsymbol{\Gamma}^{obs} = \boldsymbol{\gamma}^{obs}|\mathbf{Z} = \mathbf{z}, \Phi) = \prod_{f=1}^{F} \mathbb{P}(\boldsymbol{\Gamma}_{obs}^{f} = \boldsymbol{\gamma}_{obs}^{f}|\mathbf{Z} = \mathbf{z}, \Phi_f)$, where $\Phi = (\Phi_1, \Phi_0)$, and $\Phi_f = (\boldsymbol{m}_f, \boldsymbol{u}_f)$. Therefore we can write the likelihood for $\mathbf{Z}$ and Φ as $\mathcal{L}(\mathbf{Z}, \Phi|\boldsymbol{\Gamma}_{obs} = \boldsymbol{\gamma}_{obs}) = \prod_{f=1}^{F} \mathcal{L}(\mathbf{Z}, \Phi_f|\boldsymbol{\Gamma}_{obs}^{f} = \boldsymbol{\gamma}_{obs}^{f})$, where

$$\mathcal{L}(\mathbf{Z}, \Phi_f|\boldsymbol{\Gamma}_{obs}^{f} = \boldsymbol{\gamma}_{obs}^{f}) = \prod_{l=1}^{L_f-1} m_{fl}^{a_{fl}^{1}(\mathbf{Z})}(1 - m_{fl})^{\sum_{h>l} a_{fh}^{1}(\mathbf{Z})} u_{fl}^{a_{fl}^{0}(\mathbf{Z})}(1 - u_{fl})^{\sum_{h>l} a_{fh}^{0}(\mathbf{Z})}, \tag{12}$$

and

$$a_{fl}^{1}(\mathbf{Z}) = \sum_{i<j} I_{obs}(\boldsymbol{\gamma}_{ij}^{f}) I(\gamma_{ij}^{f} = l) \mathbf{Z}_i^T \mathbf{Z}_j, \quad a_{fl}^{0}(\mathbf{Z}) = \sum_{i<j} I_{obs}(\boldsymbol{\gamma}_{ij}^{f}) I(\gamma_{ij}^{f} = l)(1 - \mathbf{Z}_i^T \mathbf{Z}_j). \tag{13}$$

For a given matrix of memberships $\mathbf{Z}$, the two above quantities represent the number of coreferent and non–coreferent records agreeing at level l for observed comparison f.

Although our main interest is to make inferences on the coreference matrix $\boldsymbol{\Delta}$, a fully Bayesian approach requires the specification of priors for the parameters Φ.

3.7 Prior Specification

In this section we explain the choosing of the priors for the parameters m_{fl} and u_{fl}, $l = 1, \ldots, L_f - 1$. The first parameter that we will focus on is $m_{f1} = \mathbb{P}_1(\Gamma_{ij}^{f} = 1)$, which represents the probability of observing the first level of agreement in the comparison f among coreferent records. The first level of agreement represents extreme or total agreement, so if we believe that field f contains no error, m_{f1} should be, a priori, a point mass at one, and as the error in field f increases the prior of m_{f1} should get concentrated around values further from one. For example we may expect a priori m_f to be in some interval $[\lambda_{f1}^{1}, 1]$ with probability one, for some $0 < \lambda_{f1}^{1} < 1$. If the field used to compute comparison f is believed to be fairly accurate, then the threshold λ_{f1}^{1} should be set close to one. On the other hand, the more errors a field is believed to contain, the lower λ_{f1}^{1} should be set in the prior. Therefore we can take the prior distribution for m_{f1} in general as $\text{Beta}(\alpha_{f1}^{1}, \beta_{f1}^{1})$ truncated to the interval $[\lambda_{f1}^{1}, 1]$, which we denote as $\text{TBeta}(\alpha_{f1}^{1}, \beta_{f1}^{1}, \lambda_{f1}^{1}, 1)$.

The parameter $m_{f2} = \mathbb{P}_1(\Gamma_{ij}^{f} = 2|\Gamma_{ij}^{f} > 1)$ represents the probability of observing the second level of agreement in the comparison f, among coreferent record pairs that do not have the first level of agreement. Depending on the construction of the agreement levels, and if the number of levels is greater than two, we can think of the second level of agreement as mild agreement, and therefore, if we expect the amount of error to be relatively small, m_{f2} should be concentrated around values close to one. Following a similar reasoning as for m_{f1}, we could take the prior of m_{f2} as $\text{TBeta}(\alpha_{f2}^{1}, \beta_{f2}^{1}, \lambda_{f2}^{1}, 1)$, where we can set the hyper–parameters of this distribution, specially λ_{f2}^{1}, according to our expected levels of error in the field f.

We can use a similar reasoning as for m_{f1} and m_{f2} to determine the prior distribution of the remaining parameters $m_{fl} = \mathbb{P}_1(\Gamma_{ij}^{f} = l|\Gamma_{ij}^{f} > l - 1)$, $l = 3, \ldots, L_f - 1$. In general, we can take the prior of m_{fl} as $\text{TBeta}(\alpha_{fl}^{1}, \beta_{fl}^{1}, \lambda_{fl}^{1}, 1)$, where the truncation points λ_{fl}^{1} change according to the field f, the way the agreement levels were constructed, and the amount of error expected a priori. Notice however that if a field is believed to contain large amounts of errors, it may be better to exclude it from the duplicate detection process since its inclusion can potentially harm the results (Sadinle and Fienberg, 2013, explore this issue in the multiple record linkage context). In this document we set $\alpha_{fl}^{1} = \beta_{fl}^{1} = 1$, for all fields f and levels l, this is, we take $m_{fl} \sim \text{Uniform}(\lambda_{fl}^{1}, 1)$.

The probabilities u_{fl} of agreement among non–coreferent records may have quite different distributions depending on the fields used to compute the comparisons. For instance, if a nominal field is used and it contains a highly frequent category, then the probability of agreement will be high even for non–coreferent records. On the other hand, if a field is almost a unique identifier of the entities, then the probability of agreement will be small among non–coreferent records. We therefore take the priors of the u_{fl} parameters as $\text{Beta}(\alpha_{fl}^{0}, \beta_{fl}^{0})$ without truncation, but we can still use α_{fl}^{0} and β_{fl}^{0} if prior experience is available. In this document, however, we set $\alpha_{fl}^{0} = \beta_{fl}^{0} = 1$, this is $u_{fl} \sim \text{Uniform}(0, 1)$.

3.8 Bayesian Inference via Gibbs Sampler

In this section we present a Gibbs sampler to explore the posterior of $\mathbf{Z}$ given a value of the observed comparison data $\mathbf{\Gamma}_{obs}$. We have integrated over $\boldsymbol{\pi}$ in the model of Equation (7), since this leads to a simpler Gibbs sampler. The sampler uses the parametrization for the comparison data given by Equation (12). In this case, using a Gibbs sampler, we obtain a Markov chain that explores the joint posterior of $\mathbf{Z}$ and Φ. The conditional distributions necessary to implement a Gibbs sampler are given by

$$m_{fl}|\mathbf{\Gamma}^{obs} = \boldsymbol{\gamma}^{obs}, \mathbf{Z} = \mathbf{z} \sim \text{TBeta}\Big(\alpha^1_{fl} + a^1_{fl}(\mathbf{z}), \beta^1_{fl} + \sum_{h>l} a^1_{fh}(\mathbf{z}), \lambda^1_{fl}, 1\Big),$$
$$u_{fl}|\mathbf{\Gamma}^{obs} = \boldsymbol{\gamma}^{obs}, \mathbf{Z} = \mathbf{z} \sim \text{Beta}\Big(\alpha^0_{fl} + a^0_{fl}(\mathbf{z}), \beta^0_{fl} + \sum_{h>l} a^0_{fh}(\mathbf{z})\Big),$$

for $l = 1, \ldots, L_f - 1$, and $\mathbf{Z}_i|\mathbf{Z}^{(-i)} = \mathbf{z}^{(-i)}, \mathbf{\Gamma}^{obs} = \boldsymbol{\gamma}^{obs}, \Phi \sim \text{Multinomial}\big(1, (p_{i1}, \ldots, p_{i\tilde{r}})\big)$, for $i = 1, \ldots, r$, where we obtain

$$p_{iq} \propto \big(z^{(-i)}_{+q} + \alpha\big) \exp\Big(\sum_{j \neq i} z_{jq} \Lambda_{ij}\Big), \tag{14}$$

where

$$\Lambda_{ij} = \sum_{f=1}^{F} I_{obs}(\boldsymbol{\gamma}^f_{ij}) \sum_{l=1}^{L_f - 1} \left[\log\Big(\frac{m_{fl}}{u_{fl}}\Big) I(\gamma^f_{ij} = l) + \log\Big(\frac{1 - m_{fl}}{1 - u_{fl}}\Big) \sum_{h>l} I(\gamma^f_{ij} = h)\right] \tag{15}$$

$$= \sum_{f=1}^{F} I_{obs}(\boldsymbol{\gamma}^f_{ij}) \sum_{l=1}^{L_f} \log\Big(\frac{m^*_{fl}}{u^*_{fl}}\Big) I(\gamma^f_{ij} = l). \tag{16}$$

If we only have binary comparisons and no missing data, Equation (15) becomes the composite weight often used in record linkage to declare record pairs as matches or non–matches (Fellegi and Sunter, 1969; Winkler, 1988; Jaro, 1989). Equation (16) represents the composite weight proposed by Winkler (1990) using agreement levels. Finally, we notice that the conditional distributions of the m_{fl} parameters are truncated beta, which we can sample efficiently using the method of Damien and Walker (2001).

3.9 An Illustrative Example

Table 1 presents a small toy example to illustrate different situations where different sets of records may be considered as coreferent depending on the levels of error that we believe the fields may contain. We explore the results of our duplicate detection method under different scenarios where these data could have arisen. The example uses Hispanic names since dealing with those is specially challenging. Full Hispanic names are usually composed by four pieces, two corresponding to given name, and two corresponding to family name. In practice, however, Hispanic people use different pieces depending on the context, and according to their own preferences. For example, someone named *JULIAN ANDRES* (record 1 of Table 1) could be known as *JULIAN* by his extended family, but as *ANDRES* by his friends.

Records 1, 2 and 3 in Table 1 represent an example where pairwise decisions on the coreference status of records may not be transitive. In this example, records 1, 2 and 3 agree in year, month, day, and municipality; also the names in records 2 and 3 are basically contained in the pieces of name of the first record, except for some errors, but the names in records 2 and 3 completely disagree. In this situation, any method taking pairwise decisions, or even a human taking decisions for one pair of records at a time, will most likely decide that records 1 and 2 are coreferent, as well as records 1 and 3, but will probably decide that records 2 and 3 are not coreferent. Table 1 also presents records 4 and 5, which agree in all of their information, except for month, day, and the second piece of family name, which is missing for record 5. The decision of whether to declare records 4 and 5 as coreferent will depend on the levels of error that we believe month and day may contain. Below we show how the proposed method deals with the uncertainty of these situations under different scenarios.

Let us think of two different scenarios from where the records in Table 1 could have arisen. In the first scenario, each record refers to a person who was killed during a war, and the data were reported by witnesses more than 10 years after the event occurred. In this scenario, year, month, day, and municipality in Table 1 correspond to the date and location of the killing as reported by the witnesses. Under this scenario we expect to have many reporting errors in the names of the victim and in the date and place of the killing, since different witnesses may have different memories of the victims and the events.

Table 1: Toy example to illustrate that different sets of records may be considered as coreferent in different contexts.

Record	Given Name	Family Name	Year	Month	Day	Municipality
1.	JULIAN ANDRES	RAMOS ROJAS	1985	5	29	A
2.	JULIAN	RCJAS	1985	5	29	A
3.	ANDRES	RAMCS	1985	5	29	A
4.	JOSE	FLORES CANALES	1986	10	1	B
5.	JOSE	FLORES	1986	8	5	B

In the second scenario, the records in Table 1 come from tax forms, and the information was self reported. In this case, year, month, day, and municipality in Table 1 correspond to date and place of birth. The reader may agree that in this case we may expect the levels of error in all fields to be much smaller compared to the first scenario, since it is quite unlikely for one person to misreport his/her information.

In Table 2 we show a summary of how the agreement levels are constructed for the data in our toy example. The Levenshtein edit distance between two strings measures the minimum number of deletions, insertions, or replacements that we need to transform one string into the other. We use a simple modification of the Levenshtein edit distance to account for the fact that Hispanic names may have missing pieces. Basically, if name A contains one token and name B contains two tokens, the modified Levenshtein measure between A and B corresponds to the minimum of the Levenshtein distances that compare the token of name A with each token of name B. Finally, this measure is transformed to the 0–1 interval by dividing by the maximum Levenshtein distance possibly obtained between names with the lengths of A and B. In this scale, 0 means total agreement (up to missing tokens), and 1 means extreme disagreement.

In order to implement the proposed method for duplicate detection, we need to choose the prior truncation points of the parameters m_{fl}. For the sake of simplicity, we consider that our prior beliefs about each field of information can be classified in two categories: either the field is accurate or inaccurate. If field f is accurate, we take the prior truncation points for all the parameters related to this field (all m_{fl}, $l = 1, \ldots, L_f$) as 0.95, whereas if field f is inaccurate, the prior truncation points for all m_{fl}, $l = 1, \ldots, L_f$, are set to 0.7.

For simplicity, we fix the prior truncation points for Year and Municipality parameters at 0.95 for all data collection scenarios presented here. For the remaining parameters, in the killings scenario we expect the fields to contain large amounts of error, and so the prior truncation points for all parameters are set equal to 0.7 (case 1 of Figure 1), whereas for the taxes scenario, the prior truncation points are all set equal to 0.95 since a priori we expect errors to be rare (case 4 of Figure 1). We also explore two intermediate cases that fall between the previous two extreme scenarios. In the first one, we expect day and month to be pretty accurate, but given and family names to be inaccurate (case 2 of Figure 1). In the second intermediate scenario, given and family names are considered to be accurate, but day and month are considered to be inaccurate (case 3 of Figure 1).

For each set of priors, using the comparison data obtained from the records in Table 1, we run 10,000 iterations of the Gibbs sampler presented in Section 3.8, and in each case 1,000 iterations are discarded as burn–in. Figure 1 presents the posterior frequencies of the six most frequent partitions across all scenarios. Although a file with five records can be partitioned in 52 ways (5th Bell number), the six partitions presented in Figure 1 concentrate at least 99% of the posterior probability in each case. If a partition of $\{a, b, c, d, e\}$ groups records a and b together, c and d together, and e alone, it is denoted $ab/cd/e$.

Table 2: Construction of levels of agreement for the toy example in Table 1.

		Levels of Agreement			
Field	Similarity Measure	1	2	3	4
Given Name	Modified Levenshtein	0	$(0, 0.25]$	$(0.25, 0.5]$	$(0.5, 1]$
Family Name	Modified Levenshtein	0	$(0, 0.25]$	$(0.25, 0.5]$	$(0.5, 1]$
Year	Absolute Difference	0	1	2–3	4+
Month	Absolute Difference	0	1	2–3	4+
Day	Absolute Difference	0	1–2	3–7	8+
Municipality	Binary Comparison	Agree	Disagree		

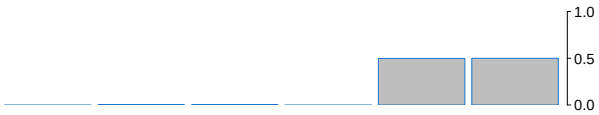
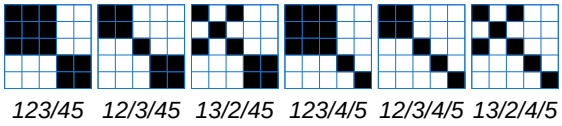
	Prior Truncation Points for $\{m_{fl}\}$		Posterior Frequencies
	Given and Family Names	Day and Month	
1.	0.70	0.70	
2.	0.70	0.95	
3.	0.95	0.70	
4.	0.95	0.95	
	Coreference Matrices		

Figure 1: Posterior distributions of the partition of the file presented in Table 1, for different sets of priors corresponding to different contexts. The six partitions presented here concentrate at least 99% of the posterior probability in each case. Prior truncation points for Year and Municipality parameters are set at 0.95 for all cases. Posterior frequencies are obtained from 9,000 iterations of a Gibbs sampler. The coreference matrices depicted here have black entries representing ones, and white entries representing zeroes.

From Figure 1 we can see that for case 1, this is, when given and family names, and day and month are inaccurate, the posterior distribution is concentrated in partitions 123/45 and 123/4/5, and from these two, it assigns more probability to 123/45. These results are reasonable given that our priors indicated that the fields were potentially inaccurate, and therefore the disagreements between fields are not taken as strong evidence of the records being non–coreferent. In case 2, we present a scenario where given and family names are still thought to be inaccurate, but day and month are believed to be accurate, and therefore we can see that in this case the posterior gets completely concentrated in the partition 123/4/5, this is, now disagreements in day and month become important for distinguishing non–coreferent records, and therefore records 4 and 5 are probably non–coreferent. In case 3, given and family names are thought to be accurate, whereas day and month to be inaccurate. In this case, the partitions where records 2 and 3 are coreferent get probability zero, since the strong disagreements between those two records now become important. Finally, in case 4, given and family names, and day and month are thought to be accurate, and therefore the partitions where records 4 and 5 are coreferent are unlikely aposteriori, as well as the partitions where records 1, 2 and 3 are clustered together. Since records 1 and 2 strongly agree, as well as records 1 and 3, but records 2 and 3 have strong disagreements, the posterior assigns equal probability to the partitions 12/3/4/5 and 13/2/4/5, which accounts properly for the uncertainty of deciding whether records 1 and 2 are coreferent, or records 1 and 3 are coreferent.

Finally, it is important to emphasize that although in this example it seems that the priors of the m_{fl} parameters completely determine the posterior of $\mathbf{\Delta}$, from Equation (15) we can see that both the m_{fl} and u_{fl} parameters influence the evolution of the memberships $\mathbf{Z}$ in the Gibbs sampler. In particular, if these five records were contained in a larger file, the resolution of their coreference status would depend on the distribution of the comparison data for the complete file, since for instance the distributions of the u_{fl}

parameters are heavily influenced by the observed frequencies of the corresponding levels of agreement.

3.10 Reducing the Inferential Complexity

Until now we have intended to make inference on the complete coreference matrix $\mathbf{\Delta}$, which represents a partition of the datafile. There are different ways to argue that this is both inefficient and unnecessary. First, the number of ways in which a datafile with r records can be partitioned is given by the rth Bell number (see e.g. Rota, 1964), which grows exponentially with r. For example, the number of possible partitions of a file with 10 records is 115,975, and if the file contains 15 records, the Bell number grows to 1,382,958,545. In practice, most files are much larger, and therefore exploring the complete space of partitions would be unfeasible. Similarly, each update of the record memberships $\mathbf{Z}_i$ in the Gibbs sampler requires the computation of $\tilde{r}$ probabilities given by Equation (14), which is computationally expensive. In most applications however, most record pairs refer to different entities, and therefore detecting obvious non–coreferent pairs may reduce tremendously the inferential and computational complexity of the problem.

Blocking is a common way to declare records as non–coreferent a priori, which consists of dividing the datafile into different blocks of records according to some reliable categorical field, or combinations of them, such that records in different blocks are considered non–coreferent. For example, if a field like gender or postal code is believed to be free of error, we could a priori declare records disagreeing on that field to be non–coreferent. We heavily rely on being able to block in order to apply the method to medium or large size datafiles.

In addition to blocking, to further reduce the complexity of the inferential task, we can a priori declare pairs of records as non–coreferent whenever they strongly disagree according to some user defined criteria. For instance, in the example presented in Section 3.9, a criterion for declaring a pair as non–coreferent could be having strong disagreements in both given and family name. Another criterion could be having strong disagreements in year, month, day, and municipality. Further criteria can be created depending on the availability of additional fields. Finally, if a pair of records meet any of the established criteria, then it is declared as non–coreferent a priori.

Notice that if records i and j are declared as non–coreferent a priori, this means that we are fixing $\Delta_{ij} = 0$, which in turn can be seen as a truncation to the prior distribution of $\mathbf{\Delta}$, since now all the partitions where records i and j are grouped together get prior probability zero. The Gibbs sampler presented in Section 3.8 can be easily modified to take into account that some Δ_{ij}'s are fixed as zero, and therefore only the remaining pairs enter the duplicate detection process. Let us denote $\mathcal{C}$ the set of candidate pairs for duplicate detection, this is, the pairs for which Δ_{ij} is not fixed as zero a priori. Notice that, $\Delta_{ij} = 0$ means $\mathbf{Z}_i^T \mathbf{Z}_j = 0$, and therefore we can write

$$a_{fl}^1(\mathbf{z}) = \sum_{(i,j)\in\mathcal{C}} I_{obs}(\boldsymbol{\gamma}_{ij}^f) I(\gamma_{ij}^f = l) \mathbf{z}_i^T \mathbf{z}_j, \tag{17}$$

$$a_{fl}^0(\mathbf{z}) = \sum_{(i,j)\notin\mathcal{C}} I_{obs}(\boldsymbol{\gamma}_{ij}^f) I(\gamma_{ij}^f = l) + \sum_{(i,j)\in\mathcal{C}} I_{obs}(\boldsymbol{\gamma}_{ij}^f) I(\gamma_{ij}^f = l)(1 - \mathbf{z}_i^T \mathbf{z}_j), \tag{18}$$

for each field f and level of agreement l. Notice that the first summand of Equation (18) does not change for different values of $\mathbf{z}$, and therefore the terms that are now constants in the $a_{fl}^0(\mathbf{z})$'s can be incorporated in the priors of the u_{fl}'s, leading to the same expressions in the Gibbs sampler for updating the u_{fl}'s, as in Section 3.8. Finally, when sampling the membership of record i conditioning on the remaining records' memberships, $\mathbf{Z}_i$ cannot be equal to $\mathbf{z}_j$ whenever $\Delta_{ij} = 0$ a priori, and therefore the truncation of the prior of $\mathbf{\Delta}$ constrains the number of memberships where i can be assigned, reducing the complexity of this step in the Gibbs sampler.

3.11 A Simulation Study

We now present a simulation study to explore the performance of the proposed methodology under different scenarios of measurement error. Peter Christen and his collaborators (Christen, 2005; Christen and Pudjijono, 2009; Christen and Vatsalan, 2013) have developed a sophisticated data generation and corruption tool to create of synthetic datasets containing various types of fields. This tool, written in Python, can include dependencies between fields, permits the generation of different types of errors, and can be easily adapted to generate additional fields that are not included in the default settings.

We now describe the characteristics of the datafiles generated here. The synthetic files contain seven fields: gender, given name, family name, postal code, phone number, age, and occupation. The fields gender

Table 3: Fields and types of errors to which they are subject in the simulation study of Section 3.11.

	Type of Error					
Field	Missing Values	Edits	OCR	Keyboard	Phonetic	Misspelling
Family Name		✓	✓	✓	✓	✓
Given Name		✓	✓	✓	✓	
Phone Number	✓	✓	✓	✓		
Postal Code	✓	✓	✓	✓		
Age Interval	✓					
Gender	✓					
Occupation	✓					

and given name are sampled jointly from a table that contains frequencies of given names per gender, and therefore popular given names appear with higher probability in the synthetic datasets. Family name and postal codes are generated independently from additional frequency tables. The three tables mentioned so far were compiled by Christen and his collaborators from public sources in Australia. Phone numbers are randomly generated following the Australian format which consists of a two–digit area code and an eight–digit number made of two blocks of four digits. The five previous fields were included in the default configuration of Christen's generator. In addition, age and occupation are jointly sampled from a contingency table that serves as an estimate of the distribution of age and occupation in Australia. The table was obtained from the webpage of the Australian Bureau of Statistics, and it contains eight categories of occupation and eight age intervals.

The generator first creates a number of original records, which are later used to create distorted duplicates. The duplicates are allocated by randomly selecting an original record, and assigning a random number of duplicates to it. The number of duplicates is generated according to a Poisson(1) truncated to the interval $[1, 5]$. Each duplicate contains a fixed number of errors distributed randomly among the different fields, but each field contains maximum two errors. The types of errors are selected at random from a set of possibilities which vary from field to field, as summarized in Table 3. Missing values means that the value of the field becomes missing. Edit errors represent random insertions, deletions, or substitutions of characters in the string. OCR errors happen typically when a document has been digitalized using optical character recognition. Keyboard errors use a keyboard layout to simulate typing errors. Phonetic errors are simulated using a list of predefined phonetic rules. Finally, misspelling errors are generated by randomly selecting one of possibly many known misspellings of a family name. For further details on the generation of these types of errors, see Christen and Pudjijono (2009) and Christen and Vatsalan (2013).

In the simulation presented here, each synthetic dataset is composed by 450 original records and 50 duplicates. To explore the performance of the method as a function of the amount of error in the datafile, we generate 100 synthetic datasets for each of four levels of error, which correspond to the number of errors per duplicate being 1, 3, 5, and 7. For each file, comparison data were created as indicated in Table 4. The record pairs having the fourth level of agreement in both given and family name were declared non–coreferent a priori, and therefore excluded from the duplicate detection process, although the frequencies of the levels of agreement among them were included in the prior, as explained in Section 3.10. The methodology is then applied to the remaining pairs under three different sets of priors. For simplicity, each set of priors have the same prior truncation point for all the m_{fl} parameters, although in a real applications the priors should be chosen carefully. The prior truncation points are 0.5, 0.8, and 0.95, which correspond to one scenario where we believe the amount of error in the file to extremely large, one where we believe it to be moderate, and one where we are overly optimistic and believe the amount of error is very limited. For each dataset, and for each set of priors, we ran 1,000 iterations of the Gibbs sampler, and discarded the first 100 as burn–in. The average runtime using an implementation in R, including the computation of the comparison data and the Gibbs sampler, was 3.4 minutes per file, on a laptop with 1.87 GHz processor and 3 GB of RAM. Before starting the complete simulation study, we obtained some longer chains for some datasets and all priors, and we could check that 900 iterations provided roughly the same frequencies of partitions as when we run longer chains.

For each datafile, and each set of priors, we obtain a sample of partitions which approximate the posterior distribution of the partition of the file. We can assess how good each partition is in terms of classifying pairs

Table 4: Construction of levels of agreement for the simulation study of Section 3.11.

		Levels of Agreement			
Field	Similarity Measure	1	2	3	4
Given Name	Levenshtein	0	$(0, 0.25]$	$(0.25, 0.5]$	$(0.5, 1]$
Family Name	Levenshtein	0	$(0, 0.25]$	$(0.25, 0.5]$	$(0.5, 1]$
Phone Number	Levenshtein	0	$(0, 0.25]$	$(0.25, 0.5]$	$(0.5, 1]$
Postal Code	Levenshtein	0	$(0, 0.25]$	$(0.25, 0.5]$	$(0.5, 1]$
Age Interval	Binary Comparison	Agree	Disagree		
Gender	Binary Comparison	Agree	Disagree		
Occupation	Binary Comparison	Agree	Disagree		

of records as coreferent and non–coreferent. If a record pair appears in the same component of a partition, then they are coreferent according to that partition. We therefore can compute the measures of *recall* and *precision* for each partition appearing in the sample. For a given partition, the measure of recall is defined as the proportion of true coreferent pairs that appear together in the same element of the partition, this is, are classified correctly by the partition. Similarly, for a given partition, the measure of precision is defined as the proportion of record pairs belonging to the same elements of the partition that are truly coreferent, this is, the proportion of pairs declared as coreferent (according to the specific partition) that are truly coreferent. These two measures are preferred for evaluating performance in duplicate detection problems, where the amount of non–coreferent pairs is large compared to the proportion of coreferent pairs, and therefore traditional measures of performance in classification, such as the misclassification rate, are misleading (Christen, 2012b, p. 165).

The results of the simulation are presented in Figure 2. Notice that for each dataset, and each set of priors, we obtain a distribution of recall and precision measures, since both of these measures are computed for each partition in the posterior. Therefore, we compute the median, the first and 99th percentile of each measure, and average over all the 100 datasets corresponding to each level of error. In Figure 2 the gray solid lines show the average of the median precisions, and the gray dashed lines show the average of the first and 99th percentiles of each measure.

We can see that when the amount of errors is small, say one and three errors per duplicate, the method works pretty well in terms of both recall and precision for the prior truncation points of 0.8 and 0.95. When the prior truncation points are equal to 0.95, and the amount of error is large, the performance of the method deteriorates in terms of recall, although the precision stays high, meaning that using this prior, any pair declared as coreferent will be truly coreferent with high probability, although it will fail detecting many coreferent pairs. When the prior truncation points are equal to 0.5, the precision of the method is low for all scenarios, which indicates that the method introduces many false coreferent pairs. Among these three priors, the best balance between recall and precision across all levels of error was obtained by setting the prior truncation points at 0.8.

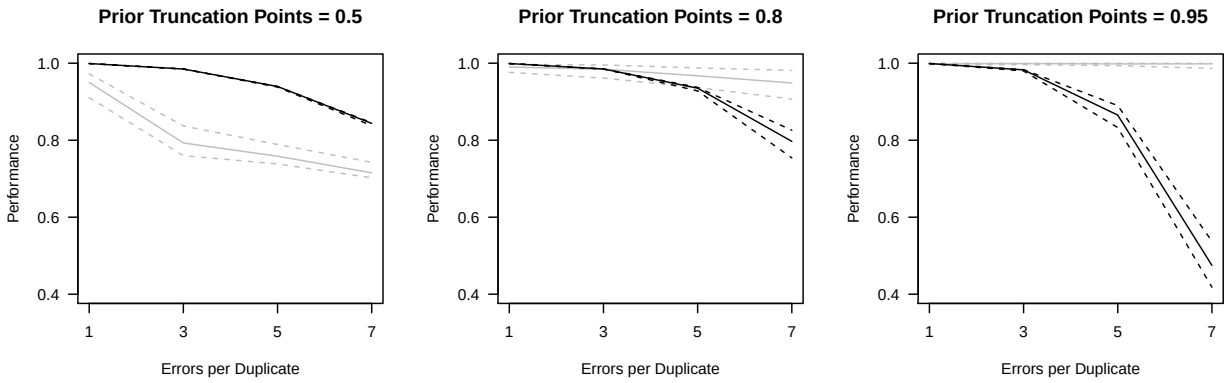


Figure 2: Recall (black lines) and precision (gray lines) for different priors in the simulation of Section 3.11.

In general, we can see that when the amount of error is small, the recall seems to be robust to prior specification, but the precision seems to be more sensitive, which indicates that when there are not many errors, it is easy to identify the truly coreferent pairs, but if our priors are overly pessimistic indicating that the amount of error is potentially much larger than what it really is, then we will end up obtaining many false coreferent pairs. On the other hand, when the amount of error is large, it seems that no prior allows us to recover all the truly coreferent pairs, and if the prior indicates that the amount of error is much smaller that what it really is, then a large proportion of the truly coreferent pairs will not be detected. The general performance can be seen as a tradeoff between recall and precision: if the priors indicate that the amount of error is too small when it is actually large, then we may end up missing too many true coreferent pairs; if the priors indicate that the amount of error is too large when it is actually small, then we may end up having too many false coreferent pairs. To have general recommendations on how to set the priors, we require further exploration of how they affect the performance of the method under different situations of error.

3.12 Extension to Joint Duplicate Detection and Record Linkage

We now propose an extension of the model presented in the previous sections to the scenario where we want to link multiple files that may contain duplicates.

As we mentioned in the Introduction, when linking multiple files it is important to bear in mind the different characteristics of each of them, including the data collection processes, and the possible errors that they may contain. For instance, the illustrative example presented in Section 3.9 shows that a pair of records may be considered coreferent under a certain data collection scenario, but non–coreferent under another.

Let D_i denote the datafile where record i belongs, this is D_i takes values in $1, \ldots, K$. The model presented in Equation (5) can be extended as follows

$$\boldsymbol{\Gamma}_{ij}|\Delta_{ij} = 1, D_i = k, D_j = l \overset{iid}{\sim} G_1^{kl}, \qquad \boldsymbol{\Gamma}_{ij}|\Delta_{ij} = 0, D_i = k, D_j = l \overset{iid}{\sim} G_0^{kl}, \tag{19}$$

where without loss of generality we can take $k \leq l$. In this model, the different distributions G^{kl} take into account different characteristics of different pairs of files. For instance, the way of comparing a pair of records may change depending on the files where the records belong, since for instance some fields might have been recorded differently in different files. Also, if files k and l are believed not to be accurate, and files k' and l' are believed to be accurate, the distributions G^{kl} and $G^{k'l'}$ will be very different, as well as the priors for the parameters in both models. Using the model presented in Equation (19) we can treat differently the duplicate detection for different files. For example the distributions G^{kk} and $G^{k'k'}$ will be different since the information in file k is believed to be inaccurate, whereas file k' is believed to contain accurate information.

In principle, the prior distribution of the coreference matrix for the combined file $\mathbf{X}$ can also be taken as in Section 3.3. Notice however that in this case we may believe that different files may contain different rates of duplicates. If this is the case, it might be important not to treat all the records as exchangeable a priori. This scenario requires considering priors different from the one considered here so far.

4 Accounting for Uncertainty from Record Linkage and Duplicate Detection in Some Bayesian Inferential Procedures

As we mentioned in Section 2.4, in principle we can propose a model for $\mathbf{X}|\phi, \boldsymbol{\Delta}$, and using Bayesian inference we can obtain $p(\phi, \boldsymbol{\Delta}|\mathbf{X}) \propto p(\mathbf{X}|\phi, \boldsymbol{\Delta})p(\phi, \boldsymbol{\Delta})$. This is the approach taken by Tancredi and Liseo (2011) in the context of capture–recapture estimation using two linked samples, and by Gutman et al. (2013) in a rather general framework for analyzing linked data from two files using Bayesian methods. Although their approach is certainly sensible, here we want to explore scenarios where the simple application of Equation (4) leads to valid inferences. The reasons for doing this is that the posterior $p(\boldsymbol{\Delta}|\mathbf{X})$ could be reused in different analyses, whereas otherwise we would have to design specific estimation methods for each different situation, and $p(\boldsymbol{\Delta}|\mathbf{X})$ can be obtained using comparison data as in Section 3. We now provide specific cases that fit under conditions 1 and 2 presented in Section 2.4.

4.1 Population Size Estimation Using Capture–Recapture Methods

A number of capture–recapture/ multiple systems estimation models have sufficient statistics that depend only on the "capture histories" of the different individuals in the files (e.g. Bishop et al., 1975; Castledine, 1981; George and Robert, 1992; Madigan and York, 1997; Fienberg et al., 1999). For example, in

a triple–systems estimation, let us denote the observed frequencies of the different capture histories as $\boldsymbol{v} = (v_{111}, v_{110}, v_{101}, v_{011}, v_{100}, v_{010}, v_{001})$, where, for example, v_{101} represents the number of individuals in files one and three, but not in file two, and the remaining elements of $\boldsymbol{v}$ are defined similarly. Many models for estimating the population size N only depend on $\boldsymbol{v}$, this is, their likelihood can be expressed as $\mathcal{L}(N|\boldsymbol{v})$. Interestingly, the vector $\boldsymbol{v}$ is a deterministic function of the coreference matrix $\boldsymbol{\Delta}$, and therefore this inferential scenario fits under Condition 1 of Section 2.4.

Let us remember from Section 2.1 that the coreference matrix $\boldsymbol{\Delta}$ for the combined file $\mathbf{X}$ can be partitioned into different submatrices. Using the notation of this section, in the case of three samples, $v_{1++} = \text{rank}(\boldsymbol{\Delta}_{11})$ denotes the number of entities represented in file 1, and we can obtain v_{+1+}, and v_{++1} in an analogous way. Similarly, $v_{11+} = \text{rank}(\boldsymbol{\Delta}_{12})$ denotes the number of entities represented in both files 1 and 2, according to the coreference matrix $\boldsymbol{\Delta}$, and we can obtain v_{1+1} and v_{+11} analogously. To obtain the vector $\boldsymbol{v}$ associated with $\boldsymbol{\Delta}$, we now only need v_{111}, which can be obtained from the following general result for K files:

Result. For K files, for a given $\boldsymbol{\Delta}$ we have that the number of entities represented in all files is given by

$$v_{11\ldots1} = \text{rank}(\boldsymbol{\Delta}_{12}\boldsymbol{\Delta}_{23}\ldots\boldsymbol{\Delta}_{(K-1)K}) = \text{rank}(\boldsymbol{\Delta}_{q_1q_2}\boldsymbol{\Delta}_{q_2q_3}\ldots\boldsymbol{\Delta}_{q_{K-1}q_K})$$

for any permutation $q_1, q_2, \ldots, q_K$ of $1, 2, \ldots, K$.

Although we omit further details on how to compute in general $\boldsymbol{v}$ from $\boldsymbol{\Delta}$, we want to emphasize that $\boldsymbol{v}$ is a deterministic function of $\boldsymbol{\Delta}$, and therefore we can make inference on N accounting for uncertainty from record linkage and duplicate detection simply as $p(N|\mathbf{X}) = \sum_{\boldsymbol{\Delta}} p(N|\boldsymbol{v}(\boldsymbol{\Delta}))p(\boldsymbol{\Delta}|\mathbf{X})$, where the posterior $p(N|\boldsymbol{v}(\boldsymbol{\Delta}))$ is obtained using $\mathcal{L}(N|\boldsymbol{v}(\boldsymbol{\Delta}))$. If the set of partitions with non–zero probability is very large, then $p(N|\mathbf{X})$ can be approximated using a sample of $\boldsymbol{\Delta}$'s.

4.2 Inference for Association Between Variables Contained in Different Files

Let us suppose that we are linking K files that do not contain duplicates, and all of them contain information on exactly the same entities, this is $r_1 = n_1 = r_2 = n_2 \cdots = r_K = n_K$. Let $\mathbf{X}_k^1$ represent a set of fields in file k that are not available in the remaining files, $k = 1, \ldots, K$. This scenario is more general than the one of Lahiri and Larsen (2005), since they focus on $K = 2$ in a regression setting. Let us suppose that we would be interested in a model with likelihood $\mathcal{L}(\phi|\mathbf{X}_1^1, \mathbf{X}_2^1, \ldots, \mathbf{X}_K^1)$ if all fields $\mathbf{X}^1 = (\mathbf{X}_1^1, \mathbf{X}_2^1, \ldots, \mathbf{X}_K^1)$ had been observed in the same file. Unfortunately, we are not able to use this model immediately since we do not know which records are coreferent in the different files. Notice that in this scenario, $\boldsymbol{\Delta}_{kl}$ is a permutation matrix, whose ith row indicates the record in file l that is coreferent with record i in file k. If we knew the true value of $\boldsymbol{\Delta}$, we could simply write the likelihood that we desire as $\mathcal{L}(\phi|\boldsymbol{\Delta}_{k1}\mathbf{X}_1^1, \boldsymbol{\Delta}_{k2}\mathbf{X}_2^1, \ldots, \mathbf{X}_k^1, \ldots, \boldsymbol{\Delta}_{kK}\mathbf{X}_K^1) := \mathcal{L}'(\phi|\mathbf{X}^1, \boldsymbol{\Delta})$, for any $k = 1, \ldots, K$, and we would be able to make traditional Bayesian inference on ϕ, obtaining a posterior $p(\phi|\mathbf{X}^1, \boldsymbol{\Delta})$. However, the uncertainty on $\boldsymbol{\Delta}$ is summarized by a posterior distribution $p(\boldsymbol{\Delta}|\mathbf{X}^2)$, perhaps $p(\boldsymbol{\Delta}|\boldsymbol{\Gamma}(\mathbf{X}^2))$ as in this thesis, where $\mathbf{X}^2$ is disjoint with $\mathbf{X}^1$. In this context, the simple formula $p(\phi|\mathbf{X}) = \sum_{\boldsymbol{\Delta}} p(\phi|\mathbf{X}^1, \boldsymbol{\Delta})p(\boldsymbol{\Delta}|\mathbf{X}^2)$ allows us to account for the uncertainty on $\boldsymbol{\Delta}$ as summarized by $p(\boldsymbol{\Delta}|\mathbf{X}^2)$. Notice that for this scenario to fit under Condition 2 of Section 2.4 we are assuming that no prior information on the relationship between the $\mathbf{X}_1^1, \mathbf{X}_2^1, \ldots, \mathbf{X}_K^1$ variables is known.

5 Future Work

The proposed work is divided in two parts. The first part includes the minimal elements that this thesis will contain, and the second part includes additional pieces that could be explored if time allows. In each part, we present the tasks following the order in which they will be performed.

- Minimal elements of the thesis.
  - Feasibility study to explore the performance of the duplicate detection method on data coming from the Commission on the Truth (CT) for El Salvador. This dataset contains 5,675 records of killings that occurred during the civil war of El Salvador. The killings were reported by witnesses and therefore many victims were reported multiple times, but no unique identifiers are available, and the levels of error in this dataset are high, which makes it a good test bed for the proposed methodology. In this context, it is important to detect records that refer to the same victim as to not overestimate the number of victims that have been reported.

- Extension of the model presented in Sections 3.2 through 3.11 to the task of joint duplicate detection and record linkage, as explained in Section 3.12. This task includes the extension of the Gibbs sampler presented in Section 3.8, simulation studies, and a comparison with the approach of Sadinle and Fienberg (2013) and Steorts et al. (2013).
- Development of the methodology for population size estimation with linked files proposed in Section 4.1. Notice that this methodology is quite general, and therefore we will focus on a couple of models for capture–recapture estimation, specifically the models of George and Robert (1992) and Madigan and York (1997), since they are some of the most basic approaches to Bayesian capture–recapture estimation. This step will include simulation studies to illustrate how the variability in the estimated population sizes vary as a function of the uncertainty in the linkage step, which in turn is a function of the amount of error in the files.
- Application of the proposed framework to provide estimates of the number of killings occurred during the civil war of El Salvador, accounting for uncertainty from record linkage and duplicate detection. In addition to the dataset from the CT for El Salvador, two NGO's collected information on killings that occurred during the war. These three sources can be used to provide capture–recapture estimates of the total number of killings.

- Additional pieces to be explored if time allows.
  - Scalability of the methodology. We plan to consider faster alternatives to the estimation procedures presented here as to allow the methodology to scale up to large datafiles.
  - Feasibility study to explore the performance of the duplicate detection method on data coming from the US census. This step will explore how the method performs on US census data corresponding to a sample of census blocks.
  - Developing the methodology presented in Section 4.2 for various scenarios. For example, the variables $\mathbf{X}_1^1, \mathbf{X}_2^1, \ldots, \mathbf{X}_K^1$ may be categorical, and we may be interested in a model of their association. Another example is when we are interested in regressing $\mathbf{X}_1^1$ on $\mathbf{X}_2^1, \ldots, \mathbf{X}_K^1$. The amount of possibilities here is large, and so we would focus on a couple of simple cases.
  - Exploration of other modeling contexts and conditions where uncertainty of record linkage and duplicate detection can be taken into account using Equation (4). This includes, for instance, extending the scenario of Section 4.2 to the more general case where the files have different sizes and different overlaps. It would also be interesting to consider the scenario where the files have duplicates, in which case measurement error models will probably play an important role.

References

Bell, R. M., Keesey, J., and Richards, T. (1994). The Urge to Merge: Linking Vital Statistics Records and Medicaid Claims. *Medical Care*, 32(10):1004–1018.

Bilenko, M., Mooney, R. J., Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003). Adaptive Name Matching in Information Integration. *IEEE Intelligent Systems*, 18(5):16–23.

Bishop, Y. M., Fienberg, S. E., and Holland, P. W. (1975). *Discrete Multivariate Analysis: Theory and Practice*. The MIT Press. Reprinted in 2007 by Springer, New York.

Castledine, B. J. (1981). A Bayesian Analysis of Multiple-Recapture Sampling for a Closed Population. *Biometrika*, 68(1):197–210.

Christen, P. (2005). Probabilistic Data Generation for Deduplication and Data Linkage. In *Proceedings of the Sixth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'05)*, pages 109–116.

Christen, P. (2008). Automatic Record Linkage using Seeded Nearest Neighbour and Support Vector Machine Classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pages 151–159. ACM.

Christen, P. (2012a). A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. *IEEE Transactions on Knowledge and Data Engineering*, 24(9):1537–1555.

Christen, P. (2012b). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer–Verlag, Berlin Heidelberg.

Christen, P. and Pudjijono, A. (2009). Accurate Synthetic Generation of Realistic Personal Information. In Theeramunkong, T., Kijsirikul, B., Cercone, N., and Ho, T.-B., editors, *Advances in Knowledge Discovery and Data Mining*, volume 5476 of *Lecture Notes in Computer Science*, pages 507–514. Springer Berlin Heidelberg.

Christen, P. and Vatsalan, D. (2013). Flexible and Extensible Generation and Corruption of Personal Data. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM 2013)*.

Cochinwala, M., Kurien, V., Lalk, G., and Shasha, D. (2001). Efficient Data Reconciliation. *Information Sciences*, 137(1–4):1–15.

Copas, J. B. and Hilton, F. J. (1990). Record Linkage: Statistical Models for Matching Computer Records. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 153(3):287–320.

Damien, P. and Walker, S. G. (2001). Sampling Truncated Normal, Beta, and Gamma Densities. *Journal of Computational and Graphical Statistics*, 10(2):206–215.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. (2007). Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16.

Fellegi, I. P. and Sunter, A. B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Fienberg, S. E., Johnson, M. S., and Junker, B. W. (1999). Classical Multilevel and Bayesian Approaches to Population Size Estimation Using Multiple Lists. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 162(3):383–405.

Fuller, W. A. (1987). *Measurement Error Models*. John Wiley & Sons, New York.

George, E. I. and Robert, C. P. (1992). Capture-Recapture Estimation Via Gibbs Sampling. *Biometrika*, 79(4):677–683.

Gutman, R., Afendulis, C. C., and Zaslavsky, A. M. (2013). A Bayesian Procedure for File Linking to Analyze End–of–Life Medical Costs. *Journal of the American Statistical Association*, 108(501):34–47.

Herzog, T. N., Scheuren, F. J., and Winkler, W. E. (2007). *Data Quality and Record Linkage Techniques*. Springer, New York.

Hogan, H. (1992). The 1990 Post-Enumeration Survey: An Overview. *The American Statistician*, 46(4):261–269.

Hogan, H. (1993). The 1990 Post–Enumeration Survey: Operations and Results. *Journal of the American Statistical Association*, 88(423):1047–1060.

Jaro, M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406):414–420.

Keener, R., Rothman, E., and Starr, N. (1987). Distributions on Partitions. *The Annals of Statistics*, 15(4):1466–1481.

Lahiri, P. and Larsen, M. D. (2005). Regression Analysis With Linked Data. *Journal of the American Statistical Association*, 100(469):222–230.

Larsen, M. D. and Rubin, D. B. (2001). Iterative Automated Record Linkage Using Mixture Models. *Journal of the American Statistical Association*, 96(453):32–41.

Madigan, D. and York, J. C. (1997). Bayesian Methods for Estimation of the Size of a Closed Population. *Biometrika*, 1(84):19–31.

McCullagh, P. (2011). *Random Permutations and Partition Models*. Springer–Verlag Berlin Heidelberg.

Méray, N., Reitsma, J. B., Ravelli, A. C., and Bonsel, G. J. (2007). Probabilistic Record Linkage is a Valid and Transparent Tool to Combine Databases Without a Patient Identification Number. *Journal of Clinical Epidemiology*, 60(9):883–891.

Rota, G.-C. (1964). The Number of Partitions of a Set. *The American Mathematical Monthly*, 71(5):498–504.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, New York.

Sadinle, M. and Fienberg, S. E. (2013). A Generalized Fellegi–Sunter Framework for Multiple Record Linkage With Application to Homicide Record Systems. *Journal of the American Statistical Association*, 108(502):385–397.

Steorts, R. C., Hall, R., and Fienberg, S. E. (2013). Bayesian Parametric Inference for High Dimensional Multiple Record Linkage. *Working Paper*.

Tancredi, A. and Liseo, B. (2011). A Hierarchical Bayesian Approach to Record Linkage and Size Population Problems. *The Annals of Applied Statistics*, 5(2B):1553–1585.

Ventura, S. L., Nugent, R., and Fuchs, E. R. H. (2013). Methods Matter: Improving USPTO Inventor Disambiguation Algorithms with Classification and Labeled Inventor Records. *Working Paper*.

Winkler, W. E. (1988). Using the EM Algorithm for Weight Computation in the Fellegi–Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods*, pages 667–671. American Statistical Association.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi–Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research Methods*, pages 354–359. American Statistical Association.

Winkler, W. E. and Thibaudeau, Y. (1991). An Application of the Fellegi–Sunter Model of Record Linkage to the 1990 U.S. Decennial Census. Statistical Research Division Technical Report 91-9, U.S. Bureau of the Census.