Journal of Privacy and Confidentiality (2015–2016) **7**, Number 1, 3–24

Probabilistic Record Linkage and Deduplication after Indexing, Blocking, and Filtering

Jared S. Murray*

Abstract.

Probabilistic record linkage, the task of merging two or more databases in the absence of a unique identifier, is a perennial and challenging problem. It is closely related to the problem of deduplicating a single database, which can be cast as linking a single database against itself. In both cases the number of possible links grows rapidly in the size of the databases under consideration, and in most applications it is necessary to first reduce the number of record pairs that will be compared.

Spurred by practical considerations, a range of methods have been developed for this task. These methods go under a variety of names, including indexing and blocking, and have seen significant development. However, methods for inferring linkage structure that account for indexing, blocking, and additional filtering steps have not seen commensurate development. In this paper we review the implications of indexing, blocking, and filtering within the popular Fellegi-Sunter framework, and propose a new model to account for particular forms of indexing and filtering.

Keywords: Record linkage, Indexing, Blocking, Fellegi-Sunter, EM algorithm, Quasi-independence.

1 Introduction

Probabilistic record linkage is the process of merging two or more databases which lack unique identifiers. The related task of detecting duplicate records in a single file can be cast as linking a file against itself, ignoring redundant comparisons. Initially developed by Newcombe et al. (1959); Newcombe and Kennedy (1962), probabilistic record linkage was mathematically formalized by Fellegi and Sunter (1969). In the ensuing decades these methods have been widely deployed, and variations on the Fellegi-Sunter framework still form the backbone of most applications of probabilistic record linkage.

Naively matching a file with N_A records to a file with N_B records requires making $N_A N_B$ comparisons as an initial step; deduplicating a single file with N records requires making $N(N-1)/2$ comparisons. Even if the comparisons themselves are relatively inexpensive to compute and the files are of moderate size, this step can be computationally

*Department of Statistics, Carnegie Mellon University, Pittsburgh, PA, mailto:jsmurray@andrew.cmu.edu. Research reported in this work was supported by the National Science Foundation under grant numbers SES-1130706 and DMS-1043903 and by the National Institute of Standards and Technology under grant number 426-47-02A. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

prohibitive. In most practical applications of probabilistic record linkage it is necessary to eliminate a large number of record pairs from consideration, without making a full comparison of the two records, a process known as *indexing* or *blocking*. Storage and other considerations often lead to an additional *filtering* step, where record pairs that are extremely unlikely to be true matches are discarded after a complete or (nearly complete) comparison has been made.

As the size of the files under consideration has increased, the development of strategies for indexing, blocking, and filtering has outstripped the capacity of models to account for them. Using the popular Fellegi-Sunter framework as a guide, this paper discusses the implications of these strategies on subsequent modeling and inference of linkage structure. We propose extensions that provide more relevant and accurate error estimates.

This paper proceeds as follows: Section 2 reviews the mathematical formulation of probabilistic record linkage and the Fellegi and Sunter (1969) framework, and describes some common strategies for reducing the number of record pairs. Section 3 discusses the modeling and inferential implications of these strategies. Section 4 develops an extension of the Fellegi and Sunter (1969) framework to account for the effects of some indexing methods. Section 5 provides illustrations on synthetic data. Section 6 concludes with discussion about extensions and the implications for other probabilistic record linkage methods.

2 Background: Probabilistic Record Linkage and Deduplication

The basic framework for linking two files is as follows: Let A and B be two databases, and let a and b generically index records in A and B. Let $a \sim b$ denote that records a and b truly correspond to the same entity, and define $M = \{(a, b) \in A \times B : a \sim b)$ and $U = \{(a, b) \in A \times B : a \not\sim b)$. The goal is to correctly classify each record pair as a match or non-match in the absence of unique identifiers. Deduplicating a single database is similar: We consider record pairs (a, a') from a single database A, with the goal of classifying each pair into matching and non-matching sets. In the remainder of the paper we use probabilistic record linkage to refer to linking two files as well as deduplicating a single file.

2.1 The Fellegi-Sunter Framework

The original method for probabilistic record linkage, which is still widely in use, was introduced by Fellegi and Sunter (1969) who formalized earlier developments by Newcombe et al. (1959); Newcombe and Kennedy (1962). See Herzog et al. (2007) for extensive review of the basic framework and extensions.

A set of fields are available in both files A and B and may be used to compare records. Often these comparisons take the form of a series of binary variables, which may indicate

direct matches on fields (do records a and b agree on gender?), sufficient agreement (is the similarity score between the two name fields greater than some threshold?), or other derived comparisons (do a and b match on month and year of birth?). Let $\gamma_{ab} = (\gamma_{ab}(1), \ldots \gamma_{ab}(q))$ be a binary vector collecting the comparisons between records a and b, taking values in $\Gamma = \{0,1\}^q$. The model for record linkage presented in Fellegi and Sunter (1969) is as follows:

$$\Pr[(a,b) \in M] = p_M \tag{1}$$

$$\Pr[\gamma_{ab} = g \mid (a,b) \in M] = \pi_{g|M} \tag{2}$$

$$\Pr[\gamma_{ab} = g \mid (a,b) \in U] = \pi_{g|U} \tag{3}$$

$$\Pr[\gamma_{ab} = g] = p_M \pi_{g|M} + (1 - p_M)\pi_{g|U}. \tag{4}$$

The probability distribution of the observed comparison vectors is a two component mixture model, where one component corresponds to true matches and the other to true non-matches. The components $\pi_{g|M}$ and $\pi_{g|U}$ are often referred to as "$m-$probabilities" and "$u-$probabilities," respectively (Winkler, 2006b).

The parameters are usually estimated via EM (Winkler, 1988). The saturated model above is typically not estimable, and it is common to assume conditional independence between comparisons so that

$$\pi_{g|M} = \prod_{j=1}^{p} \rho_{j|M}^{g(j)} (1 - \rho_{j|M})^{1-g(j)}, \quad \pi_{g|U} = \prod_{j=1}^{p} \rho_{j|U}^{g(j)} (1 - \rho_{j|U})^{1-g(j)}. \tag{5}$$

Log-linear models can be used to model conditional dependence between comparisons (see e.g., Thibaudeau (1993)). Winkler (1993) imposed additional constraints on various probabilities to improve parameter estimation.

After estimation the parameters are used to determine the linkage structure. Each record pair is classified as a match (A_1), a nonmatch (A_3), or indeterminate (A_2). Indeterminate pairs are sent out for clerical review. Fellegi and Sunter (1969) provide a decision rule that controls the following error rates:

$$\mu = P(A_1 \mid (a,b) \in U) = \sum_{g \in \Gamma} P(A_1 \mid \gamma_{ab} = g) \Pr[\gamma_{ab} = g \mid (a,b) \in U] \tag{6}$$

$$\lambda = P(A_3 \mid (a,b) \in M) = \sum_{g \in \Gamma} P(A_3 \mid \gamma_{ab} = g) \Pr[\gamma_{ab} = g \mid (a,b) \in M], \tag{7}$$

while minimizing the number of record pairs assigned to A_2. The decision rule is based on the weights

$$w_{ab} = \frac{\pi_{\gamma_{ab}|M}}{\pi_{\gamma_{ab}|U}}, \tag{8}$$

the likelihood in favor of $(a,b) \in M$. The decision rule declares (a,b) a match if $w_{ab} \geq T_\mu$, a non-match if $w_{ab} \leq T_\lambda$, and indeterminate if $T_\lambda < w_{ab} < T_\mu$. The two thresholds T_λ and T_μ are set based on specified values for μ and λ. In Fellegi and Sunter (1969) the thresholds T_λ and T_μ are determined as follows: the set of possible comparison vectors

$\gamma \in \Gamma$ is ordered such that $w_\gamma = \pi_{\gamma|M}/\pi_{\gamma|U}$ is monotonically decreasing. Index this ordered set of comparisons by $i,\ 1 \leq i \leq N_\Gamma$. Then find $1 \leq n \leq n' - 1 \leq N_\Gamma$ such that

$$\sum_{i=1}^{n-1} \pi_{\gamma_i|U} < \mu \leq \sum_{i=1}^{n} \pi_{\gamma_i|U} \tag{9}$$

$$\sum_{i=n'}^{N_\Gamma} \pi_{\gamma_i|M} \geq \lambda > \sum_{i=n'+1}^{N_\Gamma} \pi_{\gamma_i|M}. \tag{10}$$

Assume for simplicity that there exist n and n' such that $\mu = \sum_{i=1}^{n} \pi_{\gamma_i|U}$ and $\lambda = \sum_{i=n'}^{N_\Gamma} \pi_{\gamma_i|M}$ (otherwise a randomized decision rule is needed, see Fellegi and Sunter (1969) for details). Then the thresholds are given by

$$T_\mu = \frac{\pi_{\gamma_n|M}}{\pi_{\gamma_n|U}}, \quad T_\lambda = \frac{\pi_{\gamma_{n'}|M}}{\pi_{\gamma_{n'}|U}}. \tag{11}$$

2.2 Reducing the Number of Comparisons

Naively matching two files requires comparing each pair of records, which is infeasible for large files even when the comparisons are computationally inexpensive. *Indexing* techniques quickly filter out dissimilar record pairs that are extremely unlikely to be matches (Christen, 2012a, Chapter 2). Two common indexing techniques are:

- **Blocking**, which partitions records based on the values of a key like a postal code or the first initial of the last name. Blocking keys may be constructed by conjunctions of multiple keys (e.g., agreement on last initial *and* postal code). Record pairs are discarded unless they agree on the blocking key.
- **Indexing by disjunctions**, which retains record pairs that match on *one or more keys* (their disjunction). For example, we could retain only those pairs which agree on either last initial *or* postal code. More complex indexing schemes can be constructed using disjunctions of conjunctions. Indexing by disjunctions is typically carried out by doing multiple blocking passes using different keys and taking the union of all the retained pairs.[1]

It is also common to discard pairs that are not excluded by indexing, but which are unlikely to be a match. So we can add to the above list:

- **Filtering**, which discards any pairs not excluded by initial indexing steps but which are still unlikely to be a match. For example, in the case of binary comparisons we might discard any pairs (a, b) with $\gamma_{ab} = (0, 0, \ldots, 0)$ or $\sum_{j=1}^{p} \gamma_{ab}(j) \leq 1$.

[1]The terminology is not standardized in the literature; it is common for authors to ignore the distinction between what we call indexing and blocking. We follow Christen's usage here, as the distinction becomes important later.

> Filtering rules can be more complex: For example, the U.S. Census Bureau's BigMatch software filters record pairs using initial values of the m and u probabilities, $\tilde{\pi}_{g|M}$ and $\tilde{\pi}_{g|U}$, and a user provided cutoff c_0 (dropping any pairs with $\log(\tilde{\pi}_{g|M}) - \log(\tilde{\pi}_{g|U}) < c_0$) (Yancey, 2002).

Filtering can be interpreted as indexing by a particular collection of disjunctions, but unlike most indexing schemes it will generally require actually performing all or nearly all of the comparisons, negating many of the computational benefits of indexing. In Section 5 we will see that filtering can still have significant statistical value. This is especially true in the absence of high-quality keys for indexing, or in the presence of model misspecification.

Figure 1 compares blocking and indexing by disjunctions (or filtering) graphically. Observe that blocking yields a partition of records such that all links occur within and not between elements of the partition (the "blocks") (Fig. 1, left). Other indexing schemes, including indexing by disjunctions, yield "overlapping partitions" (Fig. 1, right).

Indexing by disjunctions is a way to utilize multiple keys while hedging against typographical or measurement errors that would exclude true matches. Consider the records in Table 1. Blocking on first initial of the last name captures the "Heather-Heather" pair, a likely match, but misses the "Jane-Jane" pair which is also a likely match. Blocking on the zip code captures the "Jane-Jane" pair but excludes the "Heather-Heather" pair. Either scheme probably introduces an error. But indexing by the disjunction (keeping record pairs that match on zip code *or* first initial of last name) captures both

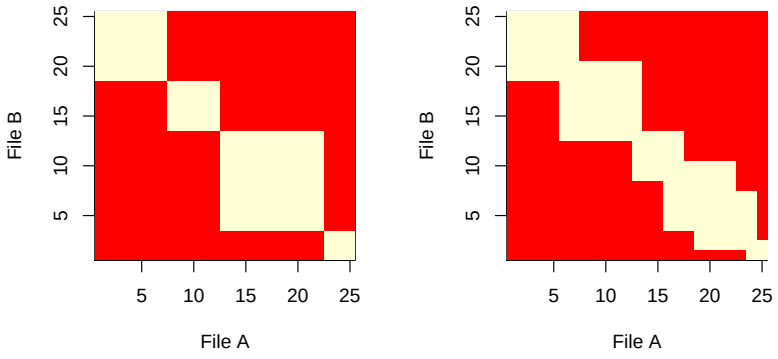

Figure 1: Comparison between blocking (left) and indexing by disjunctions (right). Record pairs in red are excluded by indexing. Blocking always partitions the *records* in each file so that records from one partition in file A are only allowed to match records from a corresponding partition in file B.

File A				File B			
First	Last	Street	Zip	First	Last	Street	Zip
Jane	Calder	123 Main St	15210	Jane	Kalder	123 Main Street	15210
Paul	Frankes	5 Birch Blvd	15232	Heather	Porter	12 Maple Ave	15236
Heather	Porter	12 Maple Ave	51236				

Table 1: Example records from two files.

pairs while excluding the unlikely "Jane-Heather" pair and both unlikely "Paul" pairs.

Indexing by disjunctions is also computationally efficient since it can be implemented by merging the results of multiple blocking queries. For these reasons it is widely used in practice. For example, Winkler et al. (2010) reports on various disjunctions used in deduplicating decennial Census records and interstate voter registration rolls, and Sadosky et al. (2015) considered indexing by disjunctions for linking records of civilian casualties in the Syria conflict. The BigMatch software developed by the U.S. Census Bureau (Yancey, 2002) was designed specifically to efficiently index by disjunctions, and also includes a subsequent filtering step.

More recent research has focused on more sophisticated methods to rapidly compute approximate dissimilarities between records using hash functions, or to infer blocking schemes using labeled matching/non-matching pairs. See e.g., Steorts et al. (2014); Christen (2012b); and Baxter et al. (2003) for reviews. We reserve discussion of these for Section 6.

3 Probabilistic Record Linkage after Indexing

Whether implemented by blocking, filtering, or some other processs, indexing creates a biased sample of record pairs by design. For model-based procedures (including Fellegi and Sunter (1969)) indexing therefore changes the interpretation of the recovered parameters. Researchers have long noted this fact (beginning at least with Fellegi and Sunter (1969) themselves; related comments appear in Jaro (1989) and Winkler (2006b)). However, the effect of indexing on subsequent modeling of record pairs and inference of linkage structure is often ignored. Using the Fellegi-Sunter framework as a guide we describe some of the implications in a simple special case, comparing the effects of traditional blocking and indexing by disjunctions (including filtering).

Let β_{ab} be an additional binary comparison, indicating whether (a, b) match on some *blocking* criterion. Similarly, let ι_{ab} be an additional binary comparison indicating agreement on some disjunction of other comparisons. We have in mind an initial blocking step that drops any record pairs with $\beta_{ab} \neq 1$, and subsequent indexing by disjunctions/filtering that drops additional record pairs with $\iota_{ab} \neq 1$.

The distinction between blocking and indexing by disjunctions or filtering is important. It would be redundant to include the comparison β_{ab} in γ_{ab}, since it is always one in the retained pairs. But when indexing by disjunctions or filtering, the compar-

isons comprising the disjunction should appear in γ_{ab}. For example, if we index by the disjunction of agreement on age and postal code, $\iota_{ab} = 1$ doesn't indicate whether there was agreement on age, postal code or both. Naturally the same argument applies when filtering. We discuss the modeling implications of blocking and of indexing by disjunctions/filtering before discussing the effect of each on the estimation of error rates and decision rules.

3.1 Modeling after Blocking

Under blocking those pairs with $\beta_{ab} \neq 1$ are treated as sure non-matches and are not used in parameter estimation. This shifts our focus from $P(\gamma_{ab})$ to $P(\gamma_{ab} \mid \beta_{ab} = 1)$. Structurally the model remains identical to (1)–(4):

$$\Pr[(a,b) \in M \mid \beta_{ab} = 1] = p_{M|\beta} \tag{12}$$

$$\Pr[\gamma_{ab} = g \mid (a,b) \in M, \beta_{ab} = 1] = \pi_{g|M,\beta} \tag{13}$$

$$\Pr[\gamma_{ab} = g \mid (a,b) \in U, \beta_{ab} = 1] = \pi_{g|U,\beta} \tag{14}$$

$$\Pr[\gamma_{ab} = g \mid \beta_{ab} = 1] = p_{M|\beta}\pi_{g|M,\beta} + (1 - p_{M|\beta})\pi_{g|U,\beta}. \tag{15}$$

The parameters, however, are not the same. In particular, we expect that:

- $p_{M|\beta} >> p_M$, provided that blocking was effective.
- $\pi_{g|M,\beta} \approx \pi_{g|M}$, since effective blocking retains (nearly) all the record pairs in M.
- $\pi_{g|U,\beta}$ may be slightly smaller than $\pi_{g|U}$ for comparison vectors g with few ones, with a commensurate increase in the conditional probability of comparison vectors with more ones. This is due to conditioning on $\beta = 1$: Given that they match on the blocking comparison, the set of retained pairs are likely to be more similar than two pairs selected at random, even if they are truly non-matches. See e.g., Jaro (1989) for further discussion. Jaro (1989) suggested estimating the $u-$probabilities using all pairs (or a randomly selected subset of all pairs), including those excluded by blocking, to mitigate bias indicated in the final bullet. This does not seem to be current practice, however. An alternative approach, which we pursue in this paper, is to explicitly acknowledge and account for the fact that inference is only valid for the subset of record pairs under consideration.

3.2 Modeling after Indexing by Disjunctions/ Filtering

When blocking is followed by another indexing step the target shifts from $P(\gamma_{ab} \mid \beta_{ab} = 1)$ to $P(\gamma_{ab} \mid \beta_{ab} = 1, \iota_{ab} = 1)$ and our model becomes

$$\Pr[(a,b) \in M \mid \beta_{ab} = 1, \iota_{ab} = 1] = p_{M|\beta,\iota} \tag{16}$$

$$\Pr[\gamma_{ab} = g \mid (a,b) \in M, \beta_{ab} = 1, \iota_{ab} = 1] = \pi_{g|M,\beta,\iota} \tag{17}$$

$$\Pr[\gamma_{ab} = g \mid (a,b) \in U, \beta_{ab} = 1, \iota_{ab} = 1] = \pi_{g|U,\beta,\iota} \tag{18}$$

$$\Pr[\gamma_{ab} = g \mid \beta_{ab} = 1, \iota_{ab} = 1] = p_{M|\beta,\iota}\pi_{g|M,\beta,\iota} + (1 - p_{M|\beta,\iota})\pi_{g|U,\beta,\iota}, \tag{19}$$

which has the same structure as (1)–(4) and (12)–(15).

The effects of indexing by disjunction or filtering are similar to those under blocking but can be more extreme. We expect that $p_{M|\beta,\iota} > p_{M|\beta} >> p_M$ when indexing or filtering and blocking are all effective. We also expect that $\pi_{g|M,\beta,\iota} \approx \pi_{g|M,\beta} \approx \pi_{g|M}$, since (nearly) all of the truly matching pairs are retained. But unlike simple applications of blocking, when indexing by disjunctions or filtering the comparison space itself can change: for example, if we index by the disjunction of exact matches on age and postal code then

$$\Pr[(a,b) \text{ disagree on age and postal code} \mid \beta_{ab} = 1, \iota_{ab} = 1] = 0,$$

so the *support* of γ_{ab} changes when conditioning on $\iota_{ab} = 1$. In general, there may be a proper subset $\Gamma_\iota \subset \Gamma$ with

$$\sum_{g \in \Gamma_\iota} \Pr[\gamma_{ab} = g \mid \beta_{ab} = 1, \iota_{ab} = 1] = 1.$$

This is an extreme version of the third bullet in Section 3.1.

Indexing by disjunctions does not necessarily change the support; for example, we might index on the disjunction of agreement on the first three digits of the postal code and agreement of age within ±5 years, but include more stringent comparisons in γ (such as matching on all postal code digits and ages within ±1 year). However, filtering restricts the support by design. Any changes in support should be explicitly reflected in subsequent modeling, which requires some modifications to the usual Fellegi-Sunter model. We discuss this further in Section 4.

3.3 Weights and Error Rates after Indexing

Unless the various bias due to indexing is specifically addressed (as proposed in Jaro (1989), for example) the estimated error rates are conditional on $\beta_{ab} = 1$ after blocking (as well as $\iota_{ab} = 1$ if blocking is followed by indexing by disjunctions or filtering). That is, under blocking we obtain estimates of

$$\mu_\beta = P(A_1 \mid (a,b) \in U, \beta_{ab} = 1) = \sum_{g \in \Gamma} P(A_1 \mid \gamma_{ab} = g, \beta_{ab} = 1)\pi_{g|U,\beta} \tag{20}$$

$$\lambda_\beta = P(A_3 \mid (a,b) \in M, \beta_{ab} = 1) = \sum_{g \in \Gamma} P(A_3 \mid \gamma_{ab} = g, \beta_{ab} = 1)\pi_{g|M,\beta}. \tag{21}$$

After blocking and indexing by disjunctions or filtering we obtain

$$\mu_{\beta,\iota} = P(A_1 \mid (a,b) \in U, \beta_{ab} = 1, \iota_{ab} = 1) = \sum_{g \in \Gamma} P(A_1 \mid \gamma_{ab} = g, \beta_{ab} = 1, \iota_{ab} = 1)\pi_{g|U,\beta,\iota} \tag{22}$$

$$\lambda_{\beta,\iota} = P(A_3 \mid (a,b) \in M, \beta_{ab} = 1, \iota_{ab} = 1) \tag{23}$$

$$= \sum_{g \in \Gamma} P(A_3 \mid \gamma_{ab} = g, \beta_{ab} = 1, \iota_{ab} = 1) \Pr[\gamma_{ab} = g \mid (a,b) \in M, \beta_{ab} = 1, \iota_{ab} = 1] \tag{24}$$

$$= \sum_{g \in \Gamma} P(A_3 \mid \gamma_{ab} = g, \beta_{ab} = 1, \iota_{ab} = 1)\pi_{g|M,\beta,\iota}. \tag{25}$$

Based on the discussion above, for most decision rules we would expect estimates of λ_β and $\lambda_{\beta,\iota}$ to be similar when indexing is functioning as intended. Both are conditional error rates and do not address error induced by indexing.

Since we expect $\pi_{g|U,\beta,\iota} > \pi_{g|U,\beta}$ for comparison vectors g with many ones, $\mu_{\beta,\iota}$ will tend to be much larger than μ_β for reasonable decision rules (which set $P(A_1 \mid (a,b) \in U, \beta_{ab} = 1, \iota_{ab} = 1)$ or $P(A_1 \mid (a,b) \in U, \beta_{ab} = 1, \iota_{ab} = 1)$ to zero for comparison vectors g that are less likely to indicate matches). But higher rates are more tolerable after additional indexing or filtering—the actual number of false matches is primarily of concern, and there are fewer total non-matching pairs under consideration. If there are n_β non-matching pairs after blocking and $k_{\beta,\iota}$ of these are excluded in a subsequent indexing/filtering step then the expected number of false matches is $\mu_\beta n_\beta$ using blocking alone and $\mu_{\beta,\iota}(n_\beta - k_{\beta,\iota})$ using blocking and indexing/filtering. Setting

$$\mu_{\beta,\iota} = \mu_\beta \frac{n_{\beta,}}{n_\beta - k_{\beta,\iota}} \tag{26}$$

provides similar control of the total number of false matches under blocking alone and blocking with additional indexing/filtering. The unknown number of true non-matching pairs n_β can be conservatively estimated by the total number of pairs remaining after blocking.

The effect of indexing by disjunctions or filtering on the weights is less obvious. Define

$$w_{g|\beta} = \frac{\Pr[\gamma_{ab} = g \mid (a,b) \in M, \beta_{ab} = 1]}{\Pr[\gamma_{ab} = g \mid (a,b) \in U, \beta_{ab} = 1]} \tag{27}$$

$$= \frac{\pi_{g|M,\beta}}{\pi_{g|U,\beta}} \tag{28}$$

$$w_{g|\beta,\iota} = \frac{\Pr[\gamma_{ab} = g \mid (a,b) \in M, \beta_{ab} = 1, \iota_{ab} = 1]}{\Pr[\gamma_{ab} = g \mid (a,b) \in U, \beta_{ab} = 1, \iota_{ab} = 1]} \tag{29}$$

$$= \frac{\pi_{g|M,\beta,\iota}}{\pi_{g|U,\beta,\iota}}. \tag{30}$$

For any comparison vector g with $\Pr[\gamma_{ab} = g \mid (a,b) \in U, \beta_{ab} = 1, \iota_{ab} = 1] > 0$,[2]

$$w_{g|\beta,\iota} = w_{g|\beta} \times \frac{\Pr[\iota_{ab} = 1 \mid \gamma_{ab} = g, (a,b) \in M, \beta_{ab} = 1]}{\Pr[\iota_{ab} = 1 \mid \gamma_{ab} = g, (a,b) \in U, \beta_{ab} = 1]} \times \frac{\Pr[\iota_{ab} = 1 \mid (a,b) \in U, \beta_{ab} = 1]}{\Pr[\iota_{ab} = 1 \mid (a,b) \in M, \beta_{ab} = 1]}. \tag{31}$$

When ι_{ab} is completely determined by γ_{ab} this simplifies to

$$w_{g|\beta,\iota} = w_{g|\beta} \times \frac{\Pr[\iota_{ab} = 1 \mid (a,b) \in U, \beta_{ab} = 1]}{\Pr[\iota_{ab} = 1 \mid (a,b) \in M, \beta_{ab} = 1]}. \tag{32}$$

This condition will hold when indexing by disjunctions of elements in γ (which includes filtering as a special case). Since the second term of (32) does not depend on g, in this case the rank order of $w_{g|\beta,\iota}$ agrees with the rank order of $w_{g|\beta}$.

In general, however, we have no such guarantee. If we assume that indexing/filtering is error-free (in that it does not exclude any truly matching pairs) we have the simple relationship

$$w_{g|\beta,\iota} = w_{g|\beta} \times \frac{\pi_{g|U,\beta}}{\pi_{g|U,\beta,\iota}}. \tag{33}$$

The second term in (33) will tend to vary across g, particularly when ι_{ab} is constructed from relaxed versions of some of the comparisons in γ_{ab}. This can alter the ranking that would be obtained from $w_{g|\beta}$ when using $w_{g|\beta,\iota}$ instead.

Similar calculations apply when comparing error rates and matching weights with and without blocking (before indexing by disjunctions/filtering). Overall it seems difficult to use the parameter estimates after indexing to make general statements about what the results would have been without indexing, even if we make generous assumptions about model specification and the errors induced by indexing. We prefer to focus explicitly on conditional versions of the parameters. When indexing by disjunctions or filtering this means our model must account for any changes in support, which requires extensions to models typically used in the Fellegi-Sunter framework.

4 Modeling Record Pairs after Indexing by Disjunctions/ Filtering

Consider the contingency table formed by the binary comparison vectors. As noted above, after filtering or indexing by disjunctions the contingency table may be *incomplete*– some cell counts are unobserved or fixed at zero (Fienberg, 1972; Bishop et al., 1975).

[2]Formally, we also require $\Pr[\iota_{ab} = 1 \mid (a,b) \in M, \beta_{ab} = 1]$ and $\Pr[\iota_{ab} = 1 \mid (a,b) \in U, \beta_{ab} = 1]$ to be nonzero. This will be the case under any practical indexing/filtering procedure.

The number of incomplete cells can be large. For example, if we index by the disjunction of two out of q total binary comparisons in γ, then $2^{(q-2)}$ of the cell counts in the table are unobserved after indexing. Subsequent filtering will generate more incomplete cells.

Incomplete cells should be treated as either structural zeros or missing data. Treating the incomplete cells as missing effectively extrapolates from the pairs remaining after indexing by disjunctions/filtering to estimate the parameters in model (12)–(15). In general, however, the estimates will be biased away from the estimates we would have gotten if we used blocking alone (data from the incomplete cells are not missing at random (Rubin, 1976)). On the other hand, treating the incomplete cells as structural zeros targets the parameters in (16)–(19) directly. The structural zero formulation is more appropriate for the following reasons:

1. In the structural zero formulation, the match/non-match probabilities, weights, error rates, and decision rule thresholds are explicitly conditional on $\iota_{ab} = 1$ and have support $\{\gamma : \gamma \in \Gamma_\iota\}$, the set of comparisons actually under consideration. This is in accordance with our discussion in the previous section and with Fellegi and Sunter (1969)'s original recommendation to explicitly specify the comparison space.

2. The proportion of true matches after blocking, $p_{M|\beta}$, is typically much smaller than $p_{M|\beta,\iota}$ because the set of excluded pairs is composed disproportionately (or entirely) of non-matching records. From a parameter estimation perspective larger values for the proportion of matches are better (see e.g., Winkler (2006b), who suggests that at least 5% of the record pairs under consideration should be matches for maximum likelihood estimates computed via EM to be reliable).

3. Under model misspecification the structural zero formulation may better approximate true values of relevant probabilities (and therefore error rates, decision rule thresholds, and matching weights). Treating the incomplete cells as structural zeros and estimating the parameters of (16)–(19) by maximum likelihood yields the parameters that best approximate $P(\gamma_{ab} \mid \beta_{ab} = 1, \iota_{ab} = 1)$ (in the Kullback-Leibler sense). In general these will be distinct from the parameters best approximating $P(\gamma_{ab} \mid \beta_{ab} = 1)$ or $P(\gamma_{ab})$, which are not of primary interest.

In the saturated model accounting for the support restriction is trivial. But the saturated model in (16)–(19) will usually not be estimable. A natural extension of the conditional independence assumption (5) to models with structural zeros is a conditional *quasi*-independence model (Goodman, 1968; Fienberg, 1970; Bishop et al., 1975):

$$\pi_{g|M,\beta,\iota} \propto \prod_{j=1}^{p} \psi^{(j)}_{g(j)|M,\beta,\iota} 1(g \in \Gamma_\iota), \quad \pi_{g|U,\beta,\iota} \propto \prod_{j=1}^{p} \psi^{(j)}_{g(j)|U,\beta,\iota} 1(g \in \Gamma_\iota). \tag{34}$$

The ψ parameters above are not identified without further constraints, but we are only concerned with the induced $m-$ and $u-$probabilities (which are identified).

The conditional quasi-independence model is straightforward to estimate via EM. Let $\{\pi^{(t)}_{g|M,\beta,\iota}, \pi^{(t)}_{g|U,\beta,\iota} : g \in \Gamma_\iota\}$ and $p^{(t)}_{M|\beta,\iota}$ be the parameters at iteration t. The EM algorithm proceeds as follows:

- (E-Step) Compute the expected cell counts for matching and non-matching pairs:

$$\tilde{n}^{(t+1)}_{g,M} = n_g s^{(t)}_g \tag{35}$$

$$\tilde{n}^{(t+1)}_{g,U} = n_g(1 - s^{(t)}_g), \tag{36}$$

  where n_g is the number of record pairs with comparison vector g and $s^{(t)}_g$ is the conditional probability that $(a, b) \in M$ given $\gamma_{ab} = g$ and the current values of the parameters:

$$s^{(t)}_g = \frac{p^{(t)}_{M|\beta,\iota}\pi^{(t)}_{g|M,\beta,\iota}}{p^{(t)}_{M|\beta,\iota}\pi^{(t)}_{g|M,\beta,\iota} + (1 - p^{(t)}_{M|\beta,\iota})\pi^{(t)}_{g|U,\beta,\iota}}. \tag{37}$$

- (M-Step 1) Set

$$p^{(t+1)}_{M|\beta,\iota} = \frac{\sum_{g\in\Gamma_\iota} \tilde{n}^{(t+1)}_{g,M}}{n}. \tag{38}$$

- (M-Step 2) Set

$$\{\pi^{(t+1)}_{g|M,\beta,\iota} : g \in \Gamma_\iota\} = \arg\max \sum_{g\in\Gamma_i} \tilde{n}^{(t+1)}_{g,M} \log\left(\pi_{g|M,\beta,\iota}\right) \tag{39}$$

$$\{\pi^{(t+1)}_{g|U,\beta,\iota} : g \in \Gamma_\iota\} = \arg\max \sum_{g\in\Gamma_i} \tilde{n}^{(t+1)}_{g,U} \log\left(\pi_{g|U,\beta,\iota}\right), \tag{40}$$

  where both maximizations are over the $|\Gamma_\iota|-$dimensional simplex.

M-step 2 is the only step that deviates from the usual EM algorithm for the conditional independence model. The maximizations must be done numerically due to the support restrictions.

A simple approach is to use (quasi-)Poisson regression, recognizing that each maximization problem above can be recast as fitting a log-linear model under quasi-independence by maximum likelihood and employing the multinomial-Poisson transform (Baker, 1994). The response vector includes *all* the cell counts, complete and incomplete, with zeros for the incomplete entries. The design matrix includes a main effect for each comparison as well as an indicator for each incomplete cell. The indicators force the estimates of incomplete cell probabilities to be zero. With a large number of cells alternative algorithms may be necessary, but this approach is feasible for binary comparisons and common values of p (less than 11 or 12). R code implementing the EM algorithm appears in Appendix 1 and is posted online.[3]

[3] `http://andrew.cmu.edu/~jsmurray/research/`

The conditional quasi-independence model can be extended along similar directions as the conditional independence model. For example, the ψ parameters in (34) can be replaced by a log-linear model with interactions. However, modeling $P(\gamma_{ab} = g \mid \beta_{ab} = 1, \iota_{ab} = 1)$ directly may confer at least some degree of robustness to the conditional quasi-independence assumption, as we will see in the example below.

5 Example: Synthetic Data (RLdata10000)

To illustrate the benefits of filtering and conditional quasi-independence models that account for it we compare the Fellegi-Sunter model under conditional quasi-independence using blocking and filtering to the standard Fellegi-Sunter model under conditional independence using blocking alone. We use a benchmark dataset (`RLdata10000`) distributed with the R package `RecordLinkage` (Borg and Sariyar, 2015). The dataset contains 9,000 distinct synthetic records of individuals. Each record has names and dates of birth generated from real German population-level data. A random sample of 1,000 of the records were appended to the dataset and corrupted. The goal is to identify these duplicate records.

Details about the exact process used to corrupt the duplicated records are not available. However, simple statistical tests indicate that the conditional independence assumption does not hold. For example, after blocking the χ^2 statistic for testing independence of agreement on first and last name among truly matching pairs is 162, with a numerically zero p-value. Therefore both models are misspecified.

The comparison vector comprises thresholded Jaro-Winkler scores for the comparisons on first and last name (Winkler, 1990) and exact matching on day, month and year of birth. The Jaro-Winkler scores are thresholded at 0.9 here. The indexing scheme begins with a traditional blocking step retaining only pairs matching on first and last initial. For the conditional quasi-independence model this is followed by a filtering step which requires that records match on at least two of the five fields (first names, last name, and day, year or month of birth). This mimics the output of programs like BigMatch (Yancey, 2002). No true matching pairs are excluded in either step. The blocking step reduces the number of pairs under consideration from $(10,000 \times 9,999)/2 = 49,995,000$ to $371,944$. After filtering, $34,896$ pairs remain.

5.1 Results

The estimates of match proportions are $\hat{p}_{M|\beta} = .0029$ and $\hat{p}_{M|\beta,\iota} = 0.032$. Both are reasonable, since the true number of matching pairs is $1,000$. The weights from the filtered and unfiltered models give the same rank order over Γ_ι. Figure 2 shows that using filtering and a conditional quasi-independence model gives improved estimates of error rates. The error rates themselves are not directly comparable, as noted in Section 4, but the filtered error rates are more relevant and better calibrated overall. For a more comparable measure we consider the relative discrepancy between nominal and actual error rates as comparison vectors are successively added to the match region of

the decision rule. The bottom panel of Figure 2 shows that the relative discrepancy is uniformly better under filtering and the conditional quasi-independence model.

We tried a variety of other thresholds for the Jaro-Winkler scores. Error rate curves appear in Figure 3. Again, the filtered error rate estimates are better calibrated. Across different thresholds the cells with highest weight typically had the same rank order with and without filtering. However, for some threshold values the rank order of cells with intermediate weights varied, so reproducing the relative discrepancy plots in the bottom panel of Figure 2 was not possible. But for the cells with highest weight the relative discrepancy was lower with filtering than using blocking alone.

6 Conclusion

We have described the effects of indexing, blocking, and filtering on subsequent inference of record linkage structure within the Fellegi-Sunter framework. Explicitly modeling the effects of indexing, and especially filtering, clarifies the interpretation of error rates and enhances their estimation (which also improves decision rules). The effects of filtering in particular will be the greatest when the files lack a small number of highly discriminative fields for use in indexing. This situation seems to be common in practice. Some of the impacts of indexing have been discussed in the literature, but modern applications of record linkage index, block or filter without making subsequent adjustments to the model for record pairs.

In related work Winkler (2006a) considered fixing a subset of highly likely/unlikely matching pairs as sure matches/non-matches during parameter estimation. This is closely related to filtering, which declares highly unlikely matching pairs as sure non-matches, but ignores them during parameter estimation. Blending the two strategies could prove fruitful. The most extreme comparison vectors could be filtered and ignored in parameter estimation, while less extreme but still very unlikely values could be fixed as sure non-matches to aid in parameter estimation.

We have focused on three simple but extremely common indexing methods: blocking, indexing by disjunctions, and filtering. A range of other techniques for indexing exist, including sophisticated approaches based on various hashing algorithms (Christen, 2012b; Steorts et al., 2014). These are perhaps best understood as fast approximations to filtering, and our discussion here applies more or less directly (especially if these indexing methods are followed by a subsequent filtering step to remove unlikely pairs that escape indexing).

The developments in this paper have applicability outside the traditional Fellegi-Sunter framework. With some relatively straightforward modifications the conditional quasi-independence model (or generalizations thereof) can be applied within in Sadinle and Fienberg (2013)'s multiple-file generalization of the Fellegi-Sunter framework. Interestingly, Sadinle and Fienberg (2013) include an example showing that blocking yields better estimates of error rates even when it is computationally feasible to make all the comparisons. But in that setting filtering is a better choice than blocking, since filtering

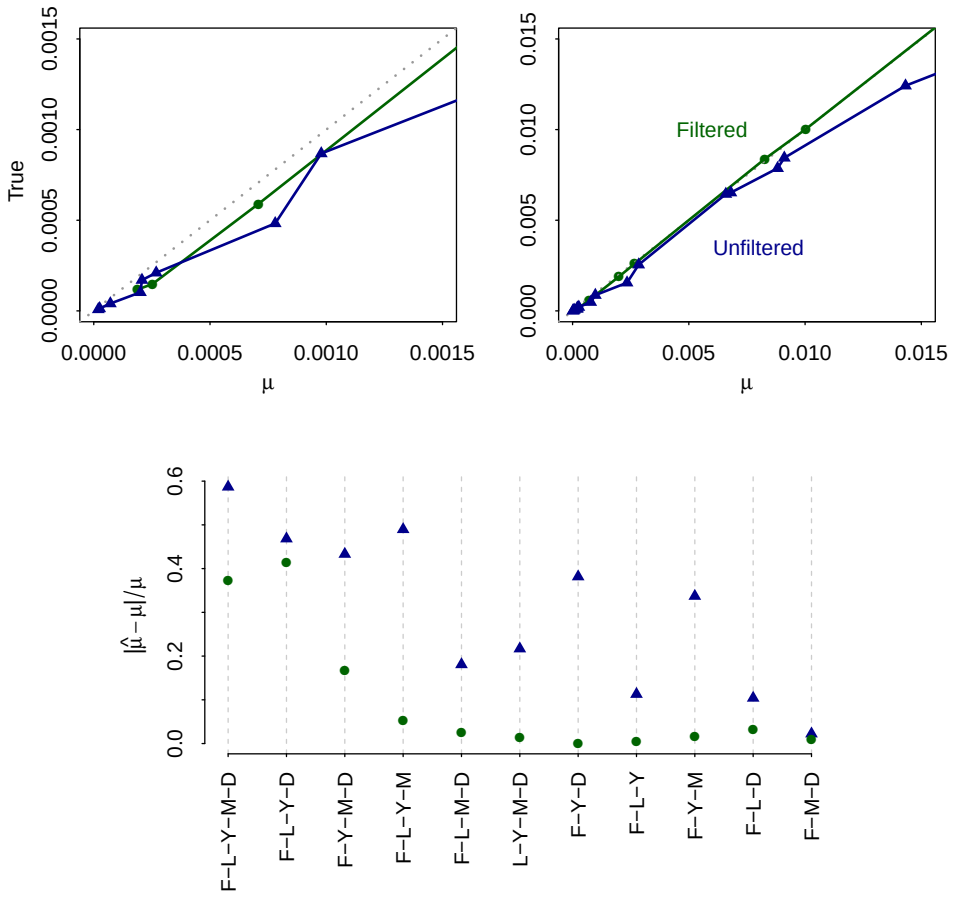

Figure 2: (Top) Estimated versus true values for the error rates $P(A_1 \mid (a,b) \in U, \beta = 1)$ (without filtering) and $P(A_1 \mid (a,b) \in U, \beta = 1, \iota = 1)$ (with filtering) when the match threshold T_μ is chosen to have nominal error rate μ. Points indicate values of μ for which the T_μ changes; intermediate values of μ rely on a randomized decision rule. (Bottom) Absolute relative discrepancy in the estimate of$P(A_1 \mid (a,b) \in U, \beta = 1)$ or $P(A_1 \mid (a,b) \in U, \beta = 1, \iota = 1)$ as comparison patterns are added to the set of declared matches. Here comparison patterns are denoted by the fields of agreement (first name, last name, year, month and day of birth), and the threshold used for string comparisons is 0.9

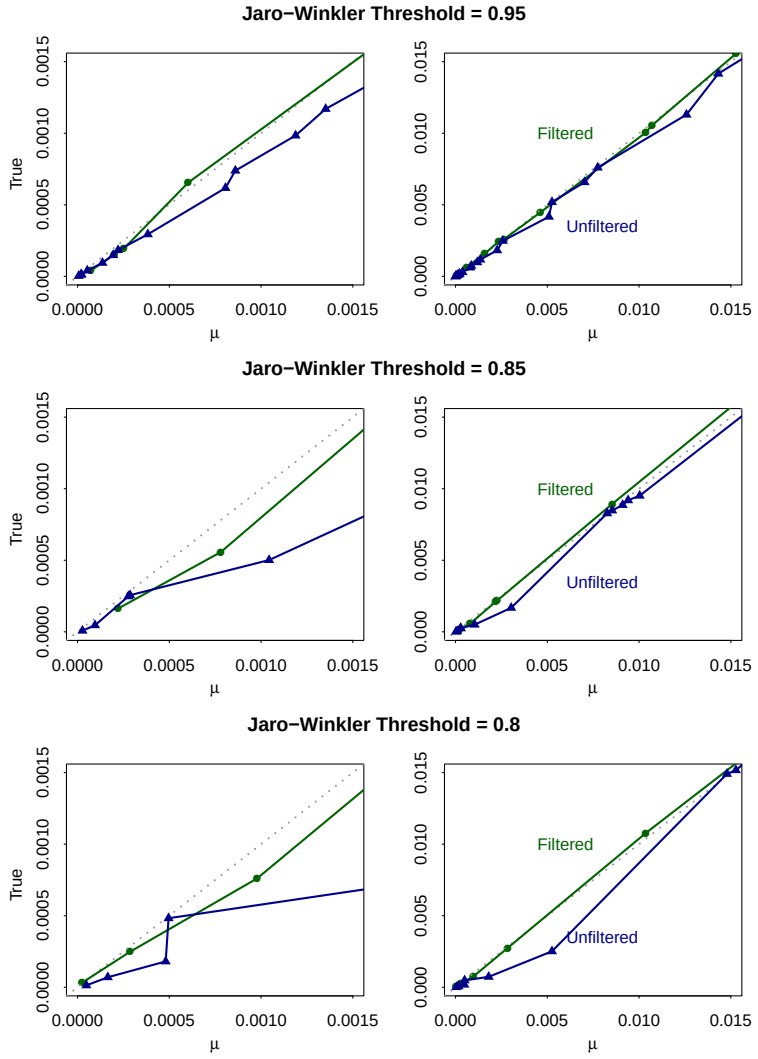

Figure 3: Estimated versus true values for the error rates $P(A_1 \mid (a,b) \in U, \beta = 1)$ (without filtering) and $P(A_1 \mid (a,b) \in U, \beta = 1, \iota = 1)$ (with filtering) when the match threshold T_μ is chosen to have nominal error rate μ. Points indicate values of μ for which the T_μ changes; intermediate values of μ rely on a randomized decision rule. Each pair of plots corresponds to a different choice of threshold for the Jaro-Winkler string comparison scores.

does not require high-quality blocking keys and will only remove pairs that are *known* to have comparison vectors which are unlikely to indicate a match. Our example above shows that filtering can accrue similar benefits in error rate estimation.

The conditional quasi-independence model introduced here is applicable within Bayesian approaches that rely on comparison vector-based likelihoods (e.g., McGlincy (2004); Larsen (2005; 2012); Sadinle (2014; 2016)). Indexing, blocking, and filtering all reduce the space of possible linkage structures that must be traversed during Markov Chain Monte Carlo, and may prove indispensable in scaling these methods to larger datasets. A challenge in this context is efficiently sampling the parameters determining the $m-$ and $u-$probabilities. The data augmentation algorithm introduced in Manrique-Vallier and Reiter (2014) is not immediately applicable, but could possibly be adapted to this purpose.

The implications of various forms of indexing for Bayesian methods that utilize full probability models for the raw data rather than comparisons are less clear (e.g., Tancredi and Liseo (2011); Gutman et al. (2013); Steorts et al. (2016); Steorts (2015)). Most of these either do not use indexing or rely on blocking, but filtering can significantly reduce the space of possible linkage structures and may play more of a role as these methods are scaled to larger problems. Indexing and filtering can complicate elicitation of joint probability models for the fields in each file. For example, when the two files have limited overlap, large and non-random subsets of records may be excluded from consideration entirely. The retained records are not exchangeable with the excluded records and prior beliefs about the complete file will not immediately transfer to the retained records.

The implications for supervised record linkage, which utilizes a set of known matching and non-matching pairs to predict unlabeled pairs, are also unclear. The biased sampling due to indexing and filtering may be largely irrelevant since the unlabeled record pairs come from a similarly biased sample. However, indexing concentrates the predictors on a subspace (e.g., Γ_ι in the context of this paper). Perhaps this dimension-reducing effect could be exploited to enhance prediction; Ventura (2015)'s blend of random forests and hierarchical clustering involves some similar ideas in this direction.

References

Baker, S. G. (1994). The multinomial-Poisson transformation. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 43(4): 495–504.
URL http://www.jstor.org/stable/2348134

Baxter, R., Christen, P., and Churches, T. (2003). A comparison of fast blocking methods for record linkage. In *ACM SIGKDD*, volume 3, 25–27.

Bishop, Y. M., Fienberg, S. E., and Holland, P. W. (1975). *Discrete multivariate analysis: theory and practice*. Springer Science & Business Media.

Borg, A. and Sariyar, M. (2015). Recordlinkage: Record linkage in r. R package version 0.4-8.
URL http://CRAN.R-project.org/package=RecordLinkage

Christen, P. (2012a). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection (Data-Centric Systems and Applications)*. Springer, 2012 edition.
URL http://amazon.com/o/ASIN/3642311636/

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

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. (1970). Quasi-independence and maximum likelihood estimation in incomplete contingency tables. *Journal of the American Statistical Association*, 65(332): 1610–1616.

— (1972). The analysis of incomplete multi-way contingency tables. *Biometrics*, 28(1): 177–202.
URL http://www.jstor.org/stable/2528967

Goodman, L. A. (1968). The analysis of cross-classified data: Independence, quasi-independence, and interactions in contingency tables with or without missing entries: R.A. Fisher memorial lecture. *Journal of the American Statistical Association*, 63(324): 1091–1131.

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, 2007 edition.
URL http://amazon.com/o/ASIN/0387695028/

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.
URL http://www.tandfonline.com/doi/abs/10.1080/01621459.198\%9.10478785

Larsen, M. D. (2005). Advances in record linkage theory: Hierarchical Bayesian record linkage theory. In *Proceedings of the Section on Survey Research Methods*.

— (2012). An experiment with hierarchical Bayesian record linkage. *arXiv preprint arXiv:1212.5203*.

Manrique-Vallier, D. and Reiter, J. P. (2014). Bayesian estimation of discrete multivariate latent structure models with structural zeros. *Journal of Computational and Graphical Statistics*, 23(4): 1061–1079.

McGlincy, M. H. (2004). A Bayesian record linkage methodology for multiple imputation of missing links. In *Proceedings of the Section on Survey Research Methods 40014008. Amer. Statist.Assoc.*.

Newcombe, H. B. and Kennedy, J. M. (1962). Record linkage: Making maximum use of the discriminating power of identifying information. *Communications of the ACM*, 5(11): 563–566.

Newcombe, H. B., Kennedy, J. M., Axford, S., and James, A. P. (1959). Automatic linkage of vital records computers can be used to extract "follow-up" statistics of families from files of routine records. *Science*, 130(3381): 954–959.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3): 581–592.

Sadinle, M. (2014). Detecting duplicates in a homicide registry using a bayesian partitioning approach. *The Annals of Applied Statistics*, 8(4): 2404–2434.

— (2016). Bayesian estimation of bipartite matchings for record linkage. *Journal of the American Statistical Association (to appear)*.

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.

Sadosky, P., Shrivastava, A., Price, M., and Steorts, R. C. (2015). Blocking methods applied to casualty records from the syrian conflict. *arXiv preprint arXiv:1510.07714*.

Steorts, R. C. (2015). Entity resolution with empirically motivated priors. *Bayesian Analysis*, 10(4): 849–875.

Steorts, R. C., Hall, R., and Fienberg, S. E. (2016). A Bayesian approach to graphical record linkage and de-duplication. *Journal of the American Statistical Association (to appear)*.

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

Tancredi, A. and Liseo, B. (2011). A hierarchical Bayesian approach to record linkage and population size problems. *Annals of Applied Statistics*, 5(2 B): 1553–1585.

Thibaudeau, Y. (1993). The discrimination power of dependency structures in record linkage. *Survey Methodology*, 19: 31–38.

Ventura, S. (2015). Large-scale classification and clustering methods with applications in record linkage. Ph.D. thesis, Carnegie Mellon University.

Winkler, W., Yancey, W., and Porter, E. (2010). Fast record linkage of very large files in support of decennial and administrative records projects. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*.

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, American Statistical Association*, volume 667, 671.

— (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*, 354–359.

— (1993). Improved decision rules in the fellegi-sunter model of record linkage. In *ASA Proceedings of Survey Research Methods Section*.

— (2006a). Automatically estimating record linkage false match rates. In *ASA Proceedings of Survey Research Methods Section*.

— (2006b). Overview of record linkage and current research directions. Technical Report Statistical Research Report Series RRC2006/02, U.S. Bureau of the Census, Washington, D.C.

Yancey, W. E. (2002). BigMatch: A program for extracting probable matches from a large file for record linkage. Technical Report Statistical Research Report Series RRC2002/01, U.S. Bureau of the Census, Washington, D.C.

1 Appendix: Example R Code

The code below is also available from `http://andrew.cmu.edu/~jsmurray/research/`.

```
# Jaro-Winkler cutoff
cutoff = 0.9

########################################################################
# Load and process the data
########################################################################
data("RLdata10000")
dat = RLdata10000

dat$fi = substr(dat$fname_c1, 1, 1)
dat$li = substr(dat$lname_c1, 1, 1)
dedup = compare.dedup(dat, blockfld=c(8,9), exclude = c(2,4,8,9),
                      strcmp = c(1,3),
                      identity=identity.RLdata10000)

pairs = dedup$pairs[,-c(1,2)]
pairs$fname_c1 = as.numeric(pairs$fname_c1>=cutoff)
pairs$lname_c1 = as.numeric(pairs$lname_c1>=cutoff)

tdf = as.data.frame(table(pairs[,-ncol(pairs)]))

keep = rowSums(sapply(tdf[,-ncol(tdf)], as.numeric)-1)>=2

trunc_tdf = tdf
trunc_tdf[!keep,ncol(tdf)] = 0
counts = trunc_tdf$Freq
n = sum(counts)

########################################################################
# Build the design matrix
########################################################################
main.eff = matrix(as.numeric(as.matrix(tdf[,1:5])), nrow=nrow(tdf))
getind = function(s, n) {rr = rep(0, n); rr[s]=1; rr }
zero.indicator = sapply(which(counts==0), getind, n=nrow(trunc_tdf))
des = data.frame(main.eff, I=zero.indicator)

########################################################################
# Control settings for the EM algorithm
########################################################################
maxiter = 1000
tol = 1e-6 # Stopping criterion

########################################################################
# Begin EM algorithm
########################################################################

# Set initial values

# p_{M \mid beta, iota}
pM = 0.1

# pi_{g\mid U, beta, iota}
# Approximately the observed frequencies, since the
# total number of matches is small (add 2 to cell counts
# to avoid issues from sampling zeros)
piU = (counts + 2)/sum(counts + 2)
piU = piU*as.numeric(keep)
piU = piU/sum(piU)

# pi_{g\mid M, beta, iota}
# Truncated conditional independence model with
# P(agree | match) = 0.95
```

```
piM = 0.95^rowSums(main.eff)*0.05^(5- rowSums(main.eff))
piM = piM*as.numeric(keep)
piM = piM/sum(piM)

for(i in 1:maxiter) {
  # E step
  cprobM = (1-pM)*piU/((1-pM)*piU + pM*piM)
  nU = counts*cprobM
  nM = counts*(1-cprob0M)
  nU[!keep] = 0
  nM[!keep] = 0

  # M step
  glm.fit.0 = glm(y~., data=data.frame(y=nU, des),
                  family="quasipoisson")
  glm.fit.1 = glm(y~., data=data.frame(y=nM, des),
                  family="quasipoisson")
  pM = sum(nM)/n

  piU.old = piU
  piM.old = piM

  g = function(fit, keep) {
    logwt = predict(fit)
    logwt = logwt - max(logwt)
    wt = as.numeric(keep)*exp(logwt)
    wt/sum(wt)
  }
  piU = g(glm.fit.0, keep)
  piM = g(glm.fit.1, keep)

  if (max(abs(log(piM/piU)[keep] - log(piM.old/piU.old)[keep])) < tol) break
}
```