Advanced Methods For Record Linkage

Article · January 1994

CITATIONS
143

READS
283

1 author:

William E. Winkler
U.S. Census Bureau
110 PUBLICATIONS **3,963** CITATIONS

SEE PROFILE

ADVANCED METHODS FOR RECORD LINKAGE 940920

William E. Winkler*, Bureau of the Census, Washington DC 20233-9100, bwinkler@census.gov

KEY WORDS: string comparator, assignment algorithm, EM algorithm, latent class

Record linkage, or computer matching, is needed for the creation and maintenance of name and address lists that support operations for and evaluations of a Year 2000 Census. This paper describes three advances. The first is an enhanced method of string comparison for dealing with typographical variations and scanning errors. It improves upon string comparators in computer science. The second is a linear assignment algorithm that can use only 0.002 as much storage as existing algorithms in operations research, requires at most an additional 0.03 increase in time, and has less of a tendency to make erroneous matching assignments than existing sparse-array algorithms because of how it deals with most arcs. The third is an expectation-maximization algorithm for estimating parameters in latent class, loglinear models of the type arising in record linkage. The associated theory and software are the only known means of dealing with general interaction patterns and allow weak use of a priori information via a generalization to the MCECM algorithm of Meng and Rubin. Models assuming that interactions are conditionally independent given the class are typically considered in biostatistics and social science.

Record linkage, or computer matching, is a means of creating, updating, and unduplicating lists that may be used in surveys. It serves as a means of linking individual records via name and address information from differing administrative files. If the files are linked using proper mathematical models, then the files can be analyzed using statistical methods such as regression and loglinear models (Scheuren and Winkler 1993).

Modern record linkage represents a collection of methods from three different disciplines: computer science, statistics, and operations research. While the foundations are from statistics, beginning with the seminal work of Newcombe (Newcombe et al. 1959, also Newcombe 1988) and Fellegi and Sunter (1969), the means of implementing the methods have primarily involved computer science. Record linkage begins with highly evolved software for parsing and standardizing names and addresses that are used in the matching. Name standardization identifies components such as first names, last names (surnames), titles, and middle initials. Address standardization locates components such as house numbers, street names, PO Boxes, and rural routes. With good standardization, effective comparison of corresponding components of information and the advanced methods described in this paper become possible. Methods from the three disciplines are needed for dealing with the three different types of problems arising in record linkage.

Because pairs of strings often exhibit typographical variation (e.g., Smith versus Smoth), the first need of record linkage is for effective string comparator functions that deal with typographical variations. While approximate string comparison has been a subject of research in computer science for many years (see survey article by Hall and Dowling 1980), the most effective ideas in the record linkage context were introduced by Jaro (1976, 1989; see also Winkler 1985, 1990). Budzinsky (1991), in an extensive review of twenty string comparison methods, concluded that the original Jaro method and the extended method due to Winkler (1990) worked second best and best, respectively.

Statistics Canada (Nuyens 1993) subsequently added string comparators based on Jaro and Winkler logic to CANLINK, Statistics Canada's matching system. This paper describes two new enhancements to the string comparators used at the Census Bureau. The first, due to McLaughlin (1993), adds logic for dealing with scanning errors (e.g., 'I' verus '1') and certain common keypunch errors (e.g., 'V' versus 'B'). The second due to Lynch and Winkler (1994) makes adjustments for pairs of long strings having a high proportion of characters in common.

The second need of record linkage is for effective means of estimating matching parameters and error rates. In addition to proving the theoretical optimality of the decision rule of Newcombe, Fellegi and Sunter (1969) showed how matching parameters could be estimated directly from available data. Their estimation methods admit closed-form solutions only if there are three matching variables and a conditional independence assumption is made. With more variables, the Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) can be used. If conditional independence is not assumed (i.e., interactions between agreements of variables such as house number, last name, and street name are allowed), then general computational algorithms (Winkler 1989) can be used. The general algorithm is an example of the MCECM algorithm of Meng and Rubin (1993). An enhancement to the basic algorithm (Winkler 1993) allows weak use of a priori information via convex constraints that restrict the solutions to subportions of the parameter space. The enhancement generalizes the MCECM algorithm.

The third need of record linkage is for a means of forcing 1-1 matching. Jaro (1989) introduced a linear sum assignment procedure (lsap) due to Burkard and Derigs (1980) as a highly effective means of eliminating many pairs that ordinarily might be clerically reviewed. With a household data source containing multiple individuals in a household, it effectively keeps the four pairs associated with father-father, mother-mother, son-son, and daughter-daughter pairs while eliminating the remaining twelve pairs associated with the household. An enhanced algorithm that uses less storage (Rowe 1987) was used during the 1990 Decennial Census (Winkler and Thibaudeau 1991). This paper describes a new algorithm (Winkler 1994a) that can use 0.002 as much storage as the Rowe algorithm and can eliminate some subtly erroneous matches that often occur in pairs of general administrative lists having only moderate overlap. In comparison with sparse-array algorithms, the new algorithm can use only 1/6 as much storage, is more than ten times as fast, and does not induce error because of the manner in which it deals with non-assigned arcs.

The next three sections describe the string comparator, the parameter-estimation algorithm, and the assignment algorithm, respectively. Each section contains examples and results that give insight into the differing methods. As the paper progresses, it increasingly indicates how the different methods affect each other but does not go as far as showing how overall matching efficacy is improved. The results of section 5 provide empirical examples of how matching efficacy is improved for three, small pairs of high quality lists. Section 5 also presents a new method for estimating error rates and compares it to the method of Belin and Rubin (1994). The discussion in section 6 describes additional uses of the new methods, some of their limitations, and future research. The final section consists of a summary and conclusion.

2. APPROXIMATE STRING COMPARISON

Dealing with typographical error can be vitally important in a record linkage context. If comparisons of pairs of strings are only done in an exact character-by-character manner, then many

matches may be lost. An extreme example is the Post Enumeration Survey (PES) (Winkler and Thibaudeau 1991, also Jaro 1989) in which, among true matches, almost 20 percent of last names and 25 percent of first names disagreed character-by-character. If matching had been performed on a character-by-character basis, then more than 30 percent of matches would have been missed by computer algorithms that were intended to delineate matches automatically. In such a situation, required manual review and (possibly) matching error would have greatly increased.

Jaro (1976, also 1989) introduced a string comparator that accounts for insertions, deletions, and transpositions. In a small study, Winkler (1985) showed that the Jaro comparator worked better than some others from computer science. In a large study, Budzinsky (1991) concluded that the comparators due to Jaro and Winkler (1990) were the best among twenty in the computer science literature. The basic Jaro algorithm is: (1) compute the string lengths, (2) find the number of common characters in the two strings, and (3) find the number of transpositions. The definition of common is that the agreeing character must be within 1/2 the length of the shorter string. The definition of transposition is that the character from one string is out of order with the correponding common character from the other string. The string comparator value (rescaled for consistency with the practice in computer science) is:

$$\text{jaro}(s1,s2) = 1/3(\ \#\text{common}/\text{str\_len1} + \#\text{common}/\text{str\_len2} + 0.5\ \#\text{transpositions}/\#\text{common}), \quad (2.1)$$

where s1 and s2 are the strings with lengths str_len1 and str_len2, respectively.

The new string comparator algorithm begins with the basic Jaro algorithm and then proceeds to three additional loops corresponding to the enhancements. Each enhancement makes use of information that is obtained from the loops prior to it.

The first enhancement due to McLaughlin (1993) assigns value 0.3 to each disagreeing but similar character. Each exact agreement gets value 1.0 and all exact agreements are located prior to searching for similar characters. Similar characters might occur because of scanning errors ('1' versus 'l') or keypunch ('V' versus 'B'). The number of common characters (#common) in equation (2.1) gets increased by 0.3 for each similar character, is denoted by #similar, and #similar is substituted for #common in the first two components of equation (2.1).

The second enhancement due to Winkler (1990) gives increased value to agreement on the beginning characters of a string. It was based on ideas from a very large empirical study by Pollock and Zamora (1984) for the Chemical Abstracts Service. The study showed that the fewest errors typically occur at the beginning of a string and the error rates by character position increase monotonically as the position moves to the right. The enhancement basically consisted of adjusting the string comparator value upward by a fixed amount if the first four characters agreed; by lesser amounts if the first three, two, or one characters agreed. The string comparator examined by Budzkinsky (1991) consisted of the Jaro comparator with only the Winkler enhancement.

The final enhancement due to Lynch and Winkler (1994) adjusts the string comparator value if the strings are longer than six characters and more than half the characters beyond the first four agree.

The final enhancement was based on detailed comparisons between versions of the comparator. The comparisons involved tens of thousands of pairs of last names, first names, and street names that did not agree on a character-by-character basis but were associated with truly matching records.

Table 2.1 illustrates the effect of the new enhanced comparators on last names, first names, and street names, respectively. If each string in a pair is less than four characters, then the Jaro and Winkler comparators return the value zero. The Jaro and Winkler comparator values are

Table 2.1. Comparison of String Comparators Using Last Names, First Names, and Street Names

Two strings		String comparator values			
		Jaro	Wink	McLa	Lynch
SHACKLEFORD	SHACKELFORD	0.970	0.982	0.982	0.989
DUNNINGHAM	CUNNIGHAM	0.896	0.896	0.896	0.931
NICHLESON	NICHULSON	0.926	0.956	0.969	0.977
JONES	JOHNSON	0.790	0.832	0.860	0.874
MASSEY	MASSIE	0.889	0.933	0.953	0.953
ABROMS	ABRAMS	0.889	0.922	0.946	0.952
HARDIN	MARTINEZ	0.722	0.722	0.722	0.774
ITMAN	SMITH	0.467	0.467	0.507	0.507
JERALDINE	GERALDINE	0.926	0.926	0.948	0.966
MARHTA	MARTHA	0.944	0.961	0.961	0.971
MICHELLE	MICHAEL	0.869	0.921	0.938	0.944
JULIES	JULIUS	0.889	0.933	0.953	0.953
TANYA	TONYA	0.867	0.880	0.916	0.933
DWAYNE	DUANE	0.822	0.840	0.873	0.896
SEAN	SUSAN	0.783	0.805	0.845	0.845
JON	JOHN	0.917	0.933	0.933	0.933
JON	JAN	0.000	0.000	0.860	0.860
BROOKHAVEN	BRROKHAVEN	0.933	0.947	0.947	0.964
BROOK HALLOW	BROOK HLLW	0.944	0.967	0.967	0.977
DECATUR	DECATIR	0.905	0.943	0.960	0.965
FITZRUREITER	FITZENREITER	0.856	0.913	0.923	0.945
HIGBEE	HIGHEE	0.889	0.922	0.922	0.932
HIGBEE	HIGVEE	0.889	0.922	0.946	0.952
LACURA	LOCURA	0.889	0.900	0.930	0.947
IOWA	IONA	0.833	0.867	0.867	0.867
1ST	IST	0.000	0.000	0.844	0.844

produced by the loop from the main production software (e.g., Winkler and Thibaudeau 1991) which is only entered if the two strings do not agree character-by-character. The return value of zero is justified because if each of the strings has three or less characters, then they necessarily disagree on at least one.

In record linkage situations, the string comparator value is used in adjusting the matching weight

associated with the comparison downward from the agreement weight toward the disagreement weight. The weighting methodology is described in the next section. Using crude statistical modeling techniques, Winkler (1990) developed downweighting functions for last names, first names, street names, and some numerical comparisons that generalized the original downweighting function introduced by Jaro.

3. PARAMETER-ESTIMATION VIA THE EM ALGORITHM

The record linkage process attempts to classify pairs in a product space **A × B** from two files A and B into M, the set of true matches, and U, the set of true nonmatches. Fellegi and Sunter (1969), making rigorous concepts introduced by Newcombe (1959), considered ratios of probabilities of the form:

$$R = P(\gamma \in \Gamma | M) / P(\gamma \in \Gamma | U) \tag{3.1}$$

where γ is an arbitrary agreement pattern in a comparison space Γ. For instance, Γ might consist of eight patterns representing simple agreement or not on the largest name component, street name, and street number. Alternatively, each $\gamma \in \Gamma$ might additionally account for the relative frequency with which specific values of name components such as "Smith", "Zabrinsky", "AAA", and "Capitol" occur.

The decision rule is given by:

If $R > UPPER$, then designate pair as a link.

If $LOWER \le R \le UPPER$, then designate pair as a possible

link and hold for clerical review. (3.2)

If $R < LOWER$, then designate pair as a nonlink.

The cutoff thresholds *UPPER* and *LOWER* are determined by a priori error bounds on false matches and false nonmatches. The three components of Rule (3.2) agree with intuition. If $\gamma \in \Gamma$ consists primarily of agreements, then it is intuitive that $\gamma \in \Gamma$ would be more likely to occur among matches than nonmatches and ratio (3.1) would be large. On the other hand, if $\gamma \in \Gamma$ consists primarily of disagreements, then ratio (3.1) would be small.

Fellegi and Sunter (1969, Theorem) showed that the decision rule is optimal in the sense that for any pair of fixed upper bounds on the rates of false matches and false nonmatches, the clerical review region is minimized over all decision rules on the same comparison space Γ. The theory holds on any subset such as pairs agreeing on a postal code, on street name, or on part of the name field. Ratio *R* or any monotonely increasing transformation of it (such as given by a logarithm) is defined as a matching weight or *total agreement weight*. In actual applications, the optimality of the decision rule (3.2) is heavily dependent on the accuracy of the estimates of the probabilities given in (3.1). The probabilities in (3.1) are called *matching parameters or matching weights*.

Fellegi and Sunter (1969, pp. 1194-1197) were the first to observe that certain parameters needed for the decision rule (3.2) could be obtained directly from observed data if certain simplifying assumptions were made. For each agreement pattern $\gamma \in \Gamma$, they considered

$$P(\gamma) = P(\gamma \mid M)\, P(M) + P(\gamma \mid U)\, P(U) \tag{3.3}$$

and noted that the proportion of pairs having representation $\gamma \in \Gamma$ could be computed directly from available data. If $\gamma \in \Gamma$ consists of a simple agree/disagree (zero/one) pattern associated with three variables satisfying the conditional independence assumption so that there exist vector constants (marginal probabilities) $m \equiv (m_1, m_2, \cdots, m_K)$ and $u \equiv (u_1, u_2, \cdots, u_K)$ such that, for all $\gamma \in \Gamma$,

$$P(\gamma \mid M) = \prod_{i=1}^{K} m_i^{\gamma^i} (1-m_i)^{(1-\gamma^i)}$$

and (3.4)

$$P(\gamma \mid U) = \prod_{i=1}^{K} u_i^{\gamma^i} (1-u_i)^{(1-\gamma^i)}.$$

then Fellegi and Sunter provide the seven solutions for the seven distinct equations associated with (3.4).

If $\gamma \in \Gamma$ represents more than three variables, then it is possible to apply general equation-solving techniques such as "the method of moments" (e.g., Hogg and Craig 1973, pp. 205-206). Because the "method of moments" has shown numerical instability in some record linkage applications (Jaro 1989, p. 417) and with general mixture distributions (Titterington et al. 1988, p. 71), maximum-likelihood-based methods such as the Expectation-Maximization (EM) algorithm (Dempster et al. 1977; also Meng and Rubin 1993) may be used.

The EM algorithm for the independence model described above has been used in a variety of record linkage situations. With each, it converged rapidly to unique limiting solutions over different starting points (Thibaudeau 1989; Winkler 1989, 1992). The major difficulty with the parameter-estimation techniques (EM or an alternative such as method of moments) is that they may yield solutions that partition the set of pairs into two sets that differ substantially from the desired sets of true matches and true nonmatches.

The basic application of the EM algorithm has been generalized in three ways. The first way involved algorithms (Thibaudeau 1989, Winkler 1989) that did not require conditional independence. Interactions between agreements on fields such as first name, last name, and house number were allowed. The second way assumed that there were three latent classes instead of two (Winkler 1992). It was stimulated by work (Smith and Newcombe 1975) showing that conditional independence does not hold because agreement on household variables (among others) are intuitively correlated. The three classes are (1) matches within households, (2) nonmatches (e.g., husband-wife) within households, and (3) nonmatches outside of households. When EM fitting using 2-class

algorithms are used, a set of pairs often divides into pairs agreeing on household information. Agreement on household variables such as last name, house number, street name, and phone overwhelm individual identifiers such as first name, age, and sex. The third way allowed convex constraints (Winkler 1992, 1993, 1994b) that restrict (predispose) solutions to subportions of the parameter space. For instance, a convex constraint might take the form:

$$P(\text{agree first, agree last} \mid \text{match}) \leq a, \tag{3.5}$$

for some $0 < a < 1$. The EM algorithm with convex constraints (Winkler 1993) generalizes the MCECM algorithm of Meng and Rubin (1993). Convex restrictions can be based on a priori knowledge of subspace regions in which modes of the likelihood yield good matching performance. In a variety of situations, having decision rules based on weights obtained from modelling interactions (with or without convex constraints) can yield accurate decision rules and reasonably accurate estimates of error rates (Winkler 1993, Armstrong and Mayda 1993). In contrast to other methods such as scoring or Newton-Raphson, the EM algorithm converges slowly but is very stable numerically (Meng and Rubin 1993).

The effects of the string comparator values and methods for adjusting weights downward from the agreement weight toward the disagreement weight are intimately tied in with the weighting methods. The basic idea of the string comparator and associated downward adjustments is to account for the partial agreement on individual fields. In the independent case, the effects of the string comparator values are much more easily understood. Details of the downward adjustment mechanism in the independent case are given in Winkler (1990). The string comparator value associated with each field is used in adjusting the weight associated with that field downward. The adjustment on each field is assumed to be independent of any other field. If the resultant weight associated with the field is negative, then the field is assumed to disagree in the string-comparator sense; otherwise, it agrees. If there are K fields, then there are 2^K agreement patterns and the total agreement weight associated with all fields is just the sum of the K string-comparator-adjusted individual-field agreement weights.

The effects of the string comparator values in the interaction case are more difficult to model and interpret because (partial) agreement in one field can no longer be assumed independent of (partial) agreement on another. During matching in the interaction (dependent) case, the agreement patterns are computed in the same manner (i.e., independently) as in the independent case. The reason for doing this is that the generalized weights divide the sets of pairs into different classes that are based on the independent patterns. Because the general weights agree much more closely with the truth, downward adjustments due to observed typographical variations in individual fields are only 1/3 what they were in the original independent case. The factor 1/3 was obtained via ad hoc means. As in the independent case, the downward adjustments are based on the estimated marginal probabilities associated with individual fields. The downward adjustments, however, are applied to the general weights. Empirically, using 1/3 the previous downward adjustment works better in all situations in which it has been used. The available data do not allow semi-rigorous modelling as in the independent case (Winkler 1990) because there are too many parameters.

The main theorem of Fellegi and Sunter is known to hold on any set of pairs. In practice, it is often

necessary to restrict results to subsets of pairs from the entire product space. In some applications, only pairs agreeing on a postal code such as U.S. ZIP code and first character of surname are considered. The restriction is referred to as *blocking*. It can reduce computation because the number of pairs is lowered by a factor of 150 with only minor (less than 5 percent) loss of true matches. By forcing 1-1 matching, the number of pairs can also be reduced. With data of the type considered in this paper and the best method of 1-1 forcing, the number of pairs can be further reduced by a factor of 9 with *no* additional loss of true matches beyond those lost via blocking. Methods of 1-1 forcing implicitly (or explicitly) can affect the weights that are used in the decision rules. The use of blocking and methods of forcing 1-1 matching can make accurate estimation of error rates much more difficult.

4. **ASSIGNMENT**

Jaro introduced a linear sum assignment procedure (lsap) to force 1-1 matching because he observed that greedy algorithms often made erroneous assignments. A greedy algorithm is one in which a record is always associated with the corresponding available record having the highest agreement weight. Subsequent records are only compared with available remaining records that have not been assigned. In the following, the two households are assumed to be the same, individuals have substantial identifying information, and the ordering is as shown.

```
HouseH1     HouseH2
husband
wife        wife
daughter    daughter
son         son
```

A greedy algorithm erroneously assigns husband1-wife2, wife1-daughter2, and daughter1-son2 whereas the lsap makes the correct assignment because it takes the highest sum of three matching weights. The original lsap algorithm of Burkard and Derigs (1980, pp. 1-11) used by Jaro (1989) could only be applied to square arrays. In some applications in which a small file representing a small geographic subregion (100 records) is matched against a large region representing an entire U.S. Postal ZIP code (20,000 records), using the typical Operations Research (OR) procedure of creating a large square array could increase storage requirements by a factor of 50 or more. To solve the initial storage difficulty, Rowe (1987) developed algorithms for rectangular arrays that were used in PES matching (Winkler and Thibaudeau 1991). Thus, the array of the above example would have dimension 50 × 20000 instead of 20000 × 20000. With square arrays, the Rowe algorithm makes identical assignments to the Burkard-Derigs algorithm.

Two difficulties arise when the Rowe algorithm is applied to general administrative lists. The first again involves size. With general lists, it is often necessary to compare all records agreeing on first character of surname and ZIP code. If the ZIP code has 60,000 individuals, then the largest array associated with agreement on first character of surname and ZIP might have dimension 5000 × 5000 which consists of 25 million numbers necessitating 100 megabytes of storage. Although it is possible to reduce array size by only considering pairs agreeing on ZIP and surname or a code of surname such as SOUNDEX, in some areas (say, involving Orientals or Hispanics) as many as 15

percent of matches would be lost.

The way the storage problem is solved is to develop an algorithm (Winkler 1994a) that only stores the highest five weights above a threshold associated with each record from each file and to keep track of the positions (rows and columns) in the original full array. All other pairs are given a default weight which is described later. The algorithm proceeds in the same graph-theoretic manner as the Burkard-Derigs-Rowe algorithm in that it uses Dijkstra's shortest augmenting path algorithm. Details differ significantly because of the mechanisms necessary for tracking the stored (compressed) array. Storage is reduced from 5000 × 5000 to 5000 × 10 because only the highest five weights and their associated column positions from the original array are stored. Based on substantial testing, computation increases only 3 percent. With most weights being replaced by defaults, much of the computation of maxima within rows or columns is eliminated. The reason overall computation increases is that extensive loops for tracking positions in arrays are needed for the small subset of stored weights.

The second difficulty involves how the weights associated with spurious agreements on fields such as first name and street name can affect the overall assignment of pairs. The choice of the default weight can solve this difficulty. The difficulty shows up as much as 0.005 of the time with original PES lists. The proportion is small because the PES has substantial overlap with Census lists and spurious agreements are not much of a problem. With general administrative lists the problem can occur much more often because the moderate overlap of the pairs of lists makes spurious agreements among nonmatches a greater problem. The following example illustrates the situation. All pairs are assumed to agree on a geographic identifier and first character of last name. Agreement and disagreement weights are added to get the total weight that is in the array used in the assignment procedure. Two records are taken from each file. reci_fj refers to record i from file j. The first record from the first file has sex miskeyed.

Table 4.1. Weights and Data Used in Assignment Examples

	last	first	hsnm	stnm	age	sex
agr_wgt	+4.0	+4.0	+2.0	+0.5	+3.0	+0.5
dis_wgt	-4.0	-4.0	-3.0	-1.0	-3.0	-4.0
rec1_f1	robert	schenck	450	main	40	F
rec2_f1	sally	schenck	450	main	40	F
rec1_f2	sally	scheck	450	main	40	F
rec2_f2	sally	smith	667	main	32	F

The array of total agreement weights is given by Array1. The rows are determined by the records from the first file and the columns are determined by the records from the second file. To obtain entry a_{11}, the first records from the respective files are compared. To obtain the total disagreement weight a_{11}, agreement and disagreement weights are added according to whether the corresponding fields in the record pair agree or disagree. The –13.0 weight in row 1 and column 2 occurs because of (spurious) agreement on street name and sex only.

$$\text{Array1} = [\ a_{ij}\ ] = \begin{matrix} 6.0 & -13.0 \\ 14.0 & -5.0 \end{matrix}$$

In this situation, $a_{11} + a_{22} = a_{12} + a_{21}$. Either 1-1 assignment is optimal. Because of the ordering, the lsap most likely would yield the erroneous assignment rec1_f1 <--> rec1_f2 and rec2_f1 <--> rec2_f2. With the new assignment algorithm, only the largest weights above a threshold are stored. If the threshold is -8.0, then the array is

$$\text{Array2} = [\ b_{ij}\ ] = \begin{matrix} 6.0 & -8.0 \\ 14.0 & -5.0 \end{matrix}$$

In this situation, the new algorithm would correctly make the assignment rec1_f1 <--> rec2_f2 and rec2_f1 <--> rec1_f2. The reason that existing sparse-array algorithms do not work is because they implicitly assign a large negative number to arcs that are not included in the graph.

$$\text{Array3} = [\ c_{ij}\ ] = \begin{matrix} 6.0 & -999.0 \\ 14.0 & -5.0 \end{matrix}$$

Using a sparse-array algorithm, the wrong assignment would *always* be made. With a number of large test decks for which true matching status is known, the original lsap, new assignment procedure, and sparse-array procedure resulted in induced error rates of 0.005, 0.000, 0.030, respectively. With pairs of lists having only moderate overlap or when different weighting methods that do not assume conditional independence (as in the above example) are used, then induced error rates with original lsap and sparse-array procedures can be much higher. In those situations, combinations of spurious agreements are more likely to occur and alternate weighting strategies may cause more error when they are combined with the original lsap.

The new assignment algorithm can reduce unnecessary erroneous assignments with general weights because some are set to −999.0 and −990.0 by the general EM program. Negative weights of large absolute value can explicitly induce error just as the sparse-array algorithms implicitly induce error. These default negative weights are designed to eliminate overflow and divide by zeros that result as certain of the estimated probabilities converge to zero. If an estimated conditional probability associated with an agreement pattern is equal to zero to 30 decimal places, then it is set equal to zero. In subsequent loops, the zero estimates necessarily remain at zero.

The weight −999.0 is a default that occurs for an agreement pattern that has estimated probability given a match equal to zero and positive probability given a nonmatch. With the general EM algorithm, estimated probabilities can converge arbitrarily close to zero; with the independent EM algorithm, all estimated probabilities must be positive. The weight -990.0 is a default that occurs when both the probability given a match and the probability given a nonmatch are zero. It is only associated with agreement patterns for which there are no observed pairs having the agreement pattern.

Algorithms that force 1-1 matching are most effective when individual files contain few duplicates.

If files contain duplicates, then such duplicates are best dealt with via special software loops (e.g., Jaro 1992, Porter 1994).

5. **RESULTS**

Results are presented in two parts. The first section provides an overall comparison of matching methods that utilize various combinations of the new and old string comparators, the new and old assignment algorithms, and the generalized interaction weighting methods and independent weighting methods. The second provides results that show how accurately error rates can be estimated using the best matching methods from the first section. Error rates are compared with rates obtained via a method of Belin and Rubin (1994) that is known to work well in a narrow range of situations (Winkler and Thibaudeau 1991, Scheuren and Winkler 1993).

5.1. Overall Comparison of Matching Methods

For comparison purposes, results are produced using three pairs of files having known matching status. The baseline matching is done under 3-class, latent class models with interactions and under independence, respectively. The 3-class models are essentially the same ones used in Winkler (1992, 1993). The interactions are (1) 8-way between last name, first name, house number, street name, phone, age, relationship to head of household, and marital status, (2) 4-way between first name, house number, phone, and sex, and (3) 2-way between last name and race. The weights associated with interaction models are referred to as *generalized weights* and other weights obtained via independence models are referred to as *independent weights*. Results are reported for error rates of 0.002, 0.005, 0.01, and 0.02, respectively. *Link*, *Nonlink*, and *Clerical (or Possible Link)* are the computer designations, respectively. *Match* and *Nonmatch* are the true statuses, respectively. The baseline results (designated by *base)* are produced using the existing lsap algorithm and the previous string comparator (see e.g., Winkler 1990) but use the newer, 3-class EM procedures for parameter estimation (Winkler 1992, 1993). The results with the new string comparator (designated *s_c*) are produced with the existing string comparator replaced by the new one. The results with the new assignment algorithm (designated *as*) use both the new string comparator and the new assignment algorithm. For comparison, results produced using the previous string comparator but with the new assignment algorithm (designated by *os_l*) are also given.

Matching efficacy improves if more pairs can be designated as links and nonlinks at fixed error rate levels. In Tables 5.1-3, computer-designated links and clerical pairs are subdivided into (true) matches and nonmatches. Only the subset of pairs produced via 1-1 assignments are considered. In producing the tables, pairs are sorted by decreasing weights. The weights vary according to the different model assumptions and string comparators used. The number of pairs above different thresholds (i.e., *UPPER* of section 3) at different link error rates (0.002, 0.005, 0.01, and 0.02) are presented. False match error rates above 2 percent are not considered because the sets of pairs above the cutoff threshold *UPPER* contain virtually all of the true matches from the entire set of pairs when error rates rise to slightly less than 2 percent. In each line under the Interaction and Independent columns, the proportion of nonmatches (among the sum of all pairs in the Link and Clerical columns) is 2 percent.

Table 5.1 Matching Results At Different Error Rates
1st Pair of Files with 4539 and 4859 records
38795 Pairs Agreeing on Block and First Char Last

Link Error Rate	Interaction		Independent	
	Link match/nonm	Clerical match/nonm	Link match/nonm	Clerical match/nonm
0.002				
base	3266/ 7	83/61	3172/ 6	242/64
s_c	2995/ 6	320/62	3176/ 6	236/64
as	3034/ 6	334/63	3176/ 6	234/64
os_l	3299/ 7	93/63	3174/ 6	242/64
0.005				
base	3312/17	37/51	3363/17	51/53
s_c	3239/17	76/51	3357/17	55/53
as	3282/17	86/52	3357/17	53/53
os_l	3354/17	38/52	3364/17	52/53
0.010				
base	3338/34	11/34	3401/34	13/36
s_c	3287/34	28/34	3396/34	16/36
as	3352/34	16/35	3396/34	14/36
os_l	3380/34	13/35	3402/34	14/36
0.020				
base	3349/68	0/ 0	3414/70	0/ 0
s_c	3315/68	0/ 0	3411/70	0/ 0
as	3368/69	0/ 0	3410/70	0/ 0
os_l	3393/69	0/ 0	3416/70	0/ 0

The results generally show that the combination of generalized weighting with the new assignment algorithm performs slightly better than the baseline with independent weighting. In all of the best situations, error levels are very low. The new string comparator produces worse results than the previous one (see e.g., Winkler 1990) and the new assignment algorithm (when combined with the new string comparator) performs slightly worse (between 0.1 and 0.01 percent) than the existing string comparator and lsap algorithm. In all situations (new or old string comparator, generalized or independent weighting), the new assignment algorithm slightly improves matching efficacy.

Table 5.2 Matching Results At Different Error Rates
2nd Pair of Files with 5022 and 5212 records
37327 Pairs Agreeing on Block and First Char Last

Link Error Rate	Interaction		Independent	
	Link match/nonm	Clerical match/nonm	Link match/nonm	Clerical match/nonm
0.002				
base	3415/ 7	102/65	3475/ 7	63/65
s_c	3308/ 7	182/64	3414/ 7	127/65
as	3326/ 7	184/65	3414/ 7	127/65
os_l	3430/ 7	107/65	3477/ 7	63/65
0.005				
base	3493/18	24/54	3503/18	35/54
s_c	3349/17	41/54	3493/18	48/54
as	3484/18	26/54	3493/18	48/54
os_l	3511/18	26/54	3505/18	36/54
0.010				
base	3501/35	16/37	3525/36	13/36
s_c	3478/35	12/38	3526/36	15/36
as	3498/35	12/37	3526/36	15/36
os_l	3519/36	18/36	3527/36	14/36
0.020				
base	3517/72	0/ 0	3538/72	0/ 0
s_c	3490/71	0/ 0	3541/72	0/ 0
as	3510/72	0/ 0	3541/72	0/ 0
os_l	3537/72	0/ 0	3541/72	0/ 0

Table 5.3 Matching Results At Different Error Rates
3rd Pair of Files with 15048 and 12072 Records
116305 Pairs Agreeing on Block and First Char Last

Link Error Rate	Interaction		Independent	
	Link match/nonm	Clerical match/nonm	Link match/nonm	Clerical match/nonm
0.002				
base	9519/19	287/181	9696/19	155/182
s_c	9462/19	338/181	9434/19	407/182
as	9418/19	410/182	9436/19	406/182
os_l	9695/19	151/182	9692/19	157/182
0.005				
base	9760/49	46/151	9792/49	59/152
s_c	9747/49	53/151	9781/49	60/152
as	9776/49	52/152	9783/49	57/152
os_l	9809/50	37/151	9791/49	58/152
0.010				
base	9784/99	22/101	9833/99	18/102
s_c	9774/99	16/101	9822/99	19/102
as	9803/99	25/102	9823/99	17/102
os_l	9828/99	18/102	9831/99	18/102
0.020				
base	9806/200	0/ 0	9851/201	0/ 0
s_c	9800/200	0/ 0	9841/201	0/ 0
as	9828/201	0/ 0	9842/201	0/ 0
os_l	9846/201	0/ 0	9849/201	0/ 0

5.2. Estimation of Error Rates

This section provides results that show how the best matching methods of the previous section can be used in estimating error rates. The basic idea is to begin with probabilities obtained for non-1-1 matching and adjust them to account (partially) for the effect of 1-1 assignment. All matching methods use the previously existing string comparator and the new assignment algorithm. Results are shown for generalized weights (Figures 1-6) and independent weights (Figures 7-12) for the same three pairs of files used in the previous section. Error rate estimates are obtained via the following steps:

1. For each agreement pattern, observe the number of pairs obtained under 1-1 matching.
2. If the weight is above 3 or below -3 (using the natural log of ratio (3.1)), then use non-1-1 probability estimates to obtain expected numbers of matches and

nonmatches (rounded to integers).

3. If the weight is between 3 and -3 and the pairs do not agree simultaneously on last name, house number and street name, use the same procedure as in 2 to get expected integer counts. If the pairs do not agree simultaneously on the three fields, then make a special adjustment to the expected counts that places an a priori upper bound on the proportion of nonmatches.

4. Use the expected integer values to obtain new estimated error rates.

The cumulations of probabilities used to get error rate estimates are via the weights used in matching. The reason that non-1-1 probabilities are used is that they yield accurate decision rules with the generalized weights (see also Winkler 1993). The true set of interactions is well approximated by the set of interactions in the model. Weights obtained under the independence model are known to yield highly inaccurate estimates of error rates in non-1-1 situations (Winkler 1989, 1992, 1993). The reason that 1-1 counts cannot be used directly in obtaining error rate estimates is covered in the discussion.

The results show that reasonably accurate estimates of the true underlying probability distributions (and thus error rates) can be obtained for matches (Figures 1-3) and nonmatches (Figures 4-6) under the generalized interaction model and for matches (Figures 7-9) and nonmatches (Figures 10-12) under the independence model. Truth is plotted against itself (45 degree line). If the estimates (jagged curve) are below the 45 degree line, they represent underestimates; if above, overestimates. Only estimated rates of 10 percent or less are displayed because at higher error rates the estimates tend to be relatively closer to the truth. Error rates higher than 10 percent are not likely to be tolerated in practical situations. Estimates with the generalized interaction weights are slightly more accurate than with independence weights. This is expected because non-1-1 error rate estimates with generalized weights are accurate (e.g., Winkler 1993) and very few (less than 2%) of true matches are lost when 1-1 matching is forced. Thus, the shapes of the curves of matches in non-1-1 and 1-1 situations are approximately the same. While the non-1-1 counts with independence weights are far from those needed for accurate estimation, it may seem suprising that the error rate estimates under the independence model are as accurate as they are. A variety of authors (Newcombe 1988, Jaro 1989, Winkler and Thibaudeau 1991) have obtained good matching results when independent weights are combined with methods of forcing 1-1 matching.

For comparison purposes, error rate estimates using the methods of this paper are compared with the method of Belin and Rubin (1994) via Figures 13-15 for independent weights and the distributions of nonmatches. With the independent weights of this paper, Belin-Rubin estimates are roughly as accurate as the independence estimates of this paper. To obtain the estimates in producing the figures, I modified Belin's software to yield estimates in a form that is consistent with the method of this paper. The current Belin-Rubin method is not intended to yield estimates for the distribution of matches and would not converge (even upon recalibration) with generalized weights.

6. **DISCUSSION**

This section provides discussion of the computing environment and software, the new string comparator, the new assignment algorithm, and methods of error rate estimation.

6.1. Computation Considerations

The computational environment primarily consisted of a Sparcstation 2. The EM software (Winkler 1994d), written in FORTRAN, and all other software, written in C, compiles and runs on other machines such as VAXes, IBM PCs, and DEC Alphas under Windows NT. The software was originally written on an IBM PC. The new assignment algorithm and string comparator are in current production software (Porter 1994) that is primarily intended for VAXes and UNIX workstations. The general EM program often requires between 5 and 20 CPU minutes for convergence. CPU time requirements of matching software are primarily dependent on the number of pairs processed and the number of fields being used in weight computations. Matching with two files of 10,000 files, 120,000 pairs (based on blocking criteria), and ten fields being compared requires approximately four minutes. The string comparison loops consume between 25 and 35 percent on the CPU time needed for matching.

The current version of the general EM software (Winkler 1994d) incorporates three new combinatorial routines that generalize combinatorial routines due to Armstrong (1992, see also Armstrong and Mayda 1993). Because the new general software allows modeling with an arbitrary number of classes and an arbitrary number of states per variable, it should facilitate modeling in non-record-linkage situations. The previous software only allowed for two or three classes and two or three value-states per variable which was appropriate for record linkage situations. It is the only latent-class software that does not assume conditional independence (Clogg 1994). Use of the combinatorial routines allows straightforward specification of the hierarchy of interaction effects and facilitates modeling and testing. While modeling is facilitated because interactions are much more easily specified, developing models still involves much trial, error, and intuition just as it does with ordinary loglinear models (Bishop, Fienberg, and Holland 1975). Early versions of the software had considerably more cumbersome methods for specifying the interactions.

6.2. String Comparator

The new string comparator is primarily designed to assist on-line searches using last name, first name, or street name. In such situations, the new comparator is believed to be superior to the old (Lynch and Winkler 1994). The reason that the new comparator performs somewhat more poorly in matching situations is that error rates with the existing methods are very low and the redundancy of extra matching fields plays a more important role than single fields in isolation. Because the new string comparator often assigns slightly higher comparator values, a few isolated true nonmatches can receive slightly higher weighting scores and observed false match rates can increase above those obtained when the original string comparators were used.

An extenuating circumstance is that the downweighting functions (Winkler 1990) have not been remodeled using the new string comparator due to the loss of the software for modeling the downweightings. I presently do not believe new downweighting functions will have more than a negligible effect on the finding of this paper that the new string comparator may slightly decrease matching efficacy when the new assignment algorithm is not used.

Presently, since there are no suitable test decks for checking scanning errors (i.e., 'I' versus '1') and some types of keypunch errors (i.e., adjacent keys 'V' versus 'B'), there has been no empirical testing of whether the associated adjustment for these types of errors helps.

The advantage of the new string comparator code is in a set of switches that can turn on or off various features. For most matching applications, the new adjustments will be turned off via switches that are in the existing source code. The string comparator code, with all the new features

activated, runs at most 40 percent slower than the previous string comparator. With the new features switched off, the speed is very close to the previous speed (Lynch and Winkler 1994).

6.3. Assignment

Other record linkage practitioners (e.g., Newcombe 1988, Jaro 1989) have observed strong evidence empirically that methods of forcing 1-1 matching can improve matching efficacy. This paper shows that certain subsets of pairs in certain types of files are much more strongly affected by the method of 1-1 forcing than others.

Based on additional tests with the existing string comparator under the independence model and the general interaction model, the new assignment algorithm always worked at least as well as the lsap that was used previously. Because the new assignment algorithm does not harm matching efficacy and always uses drastically less storage, it will always be used for matching applications.

6.4. Error Rate Estimation under the Belin-Rubin Model

The method of Belin and Rubin (1994) was designed primarily for data situations similar to PES matching. In those situations, it performed very well (Winkler and Thibaudeau 1991). Because of the weighting adjustments that were used in PES matching, the shapes of curves of matches and nonmatches were somewhat different than the corresponding shapes of the curves under the independence model used in this paper. The Belin-Rubin method is not designed to work with non-1-1 matching, for situations in which the curves of matches and nonmatches are not very well separated, or for cases in which the shapes of curves are very different from those on which Belin and Rubin originally did their modeling.

The primary advantage of the Belin-Rubin method is in its conceptual simplicity and accuracy of the estimates in those situations for which it was designed. Belin and Rubin also obtain confidence intervals via the SEM algorithm. Because of the strong simplifying assumptions, the Belin-Rubin method can be subject to bias as Belin and Rubin showed in a large simulation experiment. I have also observed some bias with independence model weights and data that is somewhat similar to the data of this paper.

6.5. Error Rate Estimation under the Model of this Paper

Using non-1-1 matching, the general interaction model of this paper provided accurate decision rules and estimates of error rates with the three pairs of data files of the results sections plus two others. Estimates were relatively more accurate than the 1-1 adjusted estimates of this paper. An example is covered in Winkler (1993).

The reason that the generalized weighting model of this paper is useful is that it can be used in a variety of non-1-1 matching situations and, with adjustments like the one of this paper, can be used in 1-1 matching situations. Because the error-rate-estimation procedure of this paper uses more information, it also may be subject to less bias than the Belin-Rubin procedure. The bias of the error-rate-estimation procedures with a variety of different types of data is a topic of future research.

6.6. Direct Estimation of Error Rates

The Belin-Rubin method of estimating error rates is logical because, with reasonable a priori assumptions on the shape of the curves of matches and nonmatches, they can provide estimates that only use the curve of weights. The curve is the one that is associated with 1-1 matching but has weights that must be estimated a priori. The next most logical estimation method is to use to the table of counts of agreement patterns obtained via 1-1 matching directly because it may allow use of more information and may not need to make a priori assumptions. There are several reasons why

such a direct use of the 1-1 counts is difficult. First, with the data of this paper, the cell counts associated with the patterns of 10 fields were nonzero for only 250 of 1024 cells. For as many as 100 of the nonzero cells, the counts were 3 or less; typically, they were 1. With the cells having nonzero counts of 3 or less, there was no apparent consistency in the proportions of matches across data files. Second, while chi-square fits of the 1-1, independence model improved by a factor of eight over the corresponding chi-square fits with the non-1-1, independence model, the chi-square values were at least a factor of six too large. The use of the asymptotic chi-square approximation with this data is not reasonable. Third, modeling of interaction effects is difficult because there appears to be many less degrees of freedom with this data than traditional statistics would suggest. For instance, with most pairs, agreement on last name appears to be almost perfectly correlated with agreement on house number, street name, and phone number. In other words, if there is agreement or disagreement on one of these variables, there is simultaneous agreement or disagreement on all the others. The race variable also has no additional discriminating power. Almost all distinguishing power is obtained from last name (alone), first name, age, marital status, relationship, and sex. At present, I have not been able to obtain good matching results with parameters that are estimated via models applied to the 1-1 counts.

7. SUMMARY AND CONCLUSION

This paper describes enhancements to a record linkage methodology that employ string comparators for dealing with strings that do not agree character-by-character, an enhanced methodology for dealing with differing, simultaneous agreements and disagreements between matching variables associated with pairs of records, and a new assignment algorithm for forcing 1-1 matching. Because of the interactions between the differing techniques, improving one method without accounting for how the method interacts with the others can actually reduce matching efficacy.

The results of this paper show that a sufficiently experienced practitioner can produce effective matching results and reasonably accurate estimates of error rates. I conclude that considerably more research is needed before the techniques can be used by naive practitioners on a large variety of administrative lists. The difficulties have the flavor of early regression analysis for which the science and art of dealing with outliers, colinearity, and other problems were not as developed as they are today. The techniques, however, can be applied by experience practitioners to a narrow range of high-quality lists such as those for evaluating Census undercount that have known matching characteristics.

REFERENCES

Armstrong, J. A. (1992), "Error Rate Estimation for Record Linkage: Some Recent Developments," in *Proceedings of the Workshop on Statistical Issues in Public Policy Analysis*, Carleton University.

Armstrong, J. B., and Mayda, J. E. (1993), "Estimation of Record Linkage Models Using Dependent Data," *Survey Methodology*, **19**, 137-147.

Belin, T. R. (1993), "Evaluation of Sources of Variation in Record Linkage through a Factorial Experiment", *Survey Methodology*, **19**, 13-29.

Belin, T. R., and Rubin, D. B. (1990), "Calibration of errors in computer matching for census undercount," *Proceedings of the Section on Government Statistics, American Statistical Association*, 124-129.

Belin, T. R., and Rubin, D. B. (1995), "A Method of Calibrating False-Match Rates in Record Linkage," *Journal of the American Statistical Association*, to appear.

Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W., (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridege, MA: MIT Press.

Budzinsky, C. D. (1991), "Automated Spelling Correction," Statistics Canada.

Burkard, R. E., and Derigs, U. (1980), *Assignment and Matching Problems: Solution Methods with FORTRAN-Programs*, New York, NY: Springer-Verlag.

Clogg, C. C. (1994), Private communication.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, **B**, **39**, 1-38.

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

Hall, P. A. V., and Dowling, G. R. (1980), "Approximate String Comparison," *Computing Surveys*, **12**, 381-402.

Hogg, R. V., and Craig, A. T. (1978), *Introduction to Mathematical Statistics*, Fourth Edition, New York, NY: J. Wiley.

Jaro, M. A. (1976), "UNIMATCH: A Record Linkage System, User's Manual," Washington, DC: U.S. Bureau of the Census.

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*, **89**, 414-420.

Jaro, M. A. (1992), "AUTOMATCH Record Linkage System," unpublished, (Available from Mathew Jaro, 14637 Locustwood Lane, Silver Spring, MD 20905, USA).

Lynch, M. P., and Winkler, W. E. (1994), "Improved String Comparator," technical report, Statistical Research Division, Washington, DC: U.S. Bureau of the Census.

McLaughlin, G. (1993), Private communication of C-string-comparison routine.

Meng, X., and Rubin, D. B. (1993), "Maximum Likelihood Via the ECM Algorithm: A General Framework," *Biometrika*, **80**, 267-278.

Newcombe, H. B. (1988), *Handbook of Record Linkage: Methods for Health and Statistical Studies, Administration, and Business*, Oxford: Oxford University Press.

Newcombe, H. B., Kennedy, J. M., Axford, S. J., and James, A. P. (1959), "Automatic Linkage of Vital Records," *Science*, **130**, 954-959.

Nuyens, C. (1993), "Generalized Record Linkage at Statistics Canada," *Proceedings of the International Conference on Establishment Surveys*, Alexandria, VA: American Statistical Association, 926-930.
Pollock, J. and Zamora, A. (1984), "Automatic Spelling Correction in Scientific and Scholarly Text," *Communications of the ACM*, **27**, 358-368.
Porter, E. (1994), "Redesigned Record Linkage Software," tech report, Statistical Research Division, Washington, DC: U.S. Bureau of the Census.
Rowe, E. (1987), Private communication of a FORTRAN lsap routine.
Rubin, D. B., and Belin, T. R. (1991), "Recent Developments in Calibrating Error Rates for Computer Matching," *Proceedings of the 1991 Census Annual Research Conference*, 657-668.
Scheuren, F., and Winkler, W. E. (1993), "Regression Analysis of Data Files that are Computer Matched," *Survey Methodology*, **19**, 39-58.
Smith, M. E., and Newcombe, H. B. (1975), "Methods of Computer Linkage of Hospital Admission-Separation Records into Cumulative Health Histories," *Methods of Information in Medicine*, **14**, 118-125.
Statistics Canada (1983), "Generalized Iterative Record Linkage System," unpublished report, Ottawa, Ontario, Canada: Systems Development Division.
Thibaudeau, Y. (1989), "Fitting Log-Linear Models When Some Dichotomous Variables are Unobservable," in *Proceedings of the Section on Statistical Computing, American Statistical Association*, 283-288.
Thibaudeau, Y. (1993), "The Discrimination Power of Dependency Structures in Record Linkage," *Survey Methodology*, **19**, 31-38.
Titterington, D. M., Smith, A. F. M., Makov, U. E. (1988), *Statistical Analysis of Finite Mixture Distributions*, New York: J. Wiley.
Winkler, W. E. (1985), "Preprocessing of Lists and String Comparison," in W. Alvey and B. Kilss, (eds.) *Record Linkage Techniques- 1985*, U.S. Internal Revenue Service, Publication 1299 (2-86), 181-187.
Winkler, W. E. (1988), "Using the EM Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 667-671.
Winkler, W. E. (1989), "Near Automatic Weight Computation in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Fifth Census Bureau Annual Research Conference*, 145-155.
Winkler, W. E. (1990), "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 354-359.
Winkler, W. E. (1991), "Error Model for Analysis of Computer Linked Files," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 472-477.
Winkler, W. E. (1992), "Comparative Analysis of Record Linkage Decision Rules," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 829-834.
Winkler, W. E. (1993), "Improved Decision Rules in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Section on Survey Research Methods, American Statistical Association*, 274-279.

Winkler, W. E. (1994a), "Improved Matching via a New Assignment Algorithm," technical report, Statistical Research Division, Washington, DC: U.S. Bureau of the Census.
Winkler, W. E. (1994b), "Improved Parameter Estimation in Record Linkage," technical report, Statistical Research Division, Washington, DC: U.S. Bureau of the Census.
Winkler, W. E. (1994c), "Matching and Record Linkage," in B. G. Cox (ed.) *Survey Methods for Businesses, Farms, and Institutions*, New York: J. Wiley, to appear.
Winkler, W. E. (1994d), "Documentation of General EM Software for Latent Class Models," technical report, Statistical Research Division, Washington, DC: U.S. Bureau of the Census.
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 Report 91/09, Washington, DC: U.S. Bureau of the Census.

Figure 1. Estimates vs Truth
Cumulative Distribution of Matches
1st Files, Interaction EM, 1−1

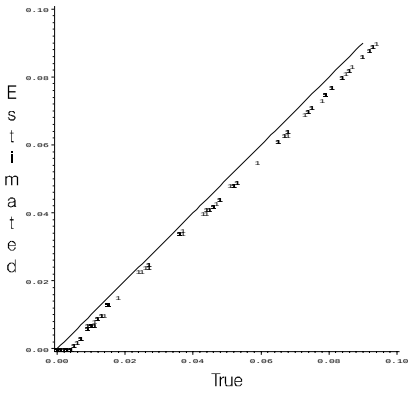

Figure 2. Estimates vs Truth
Cumulative Distribution of Matches
2nd Files, Interaction EM, 1−1

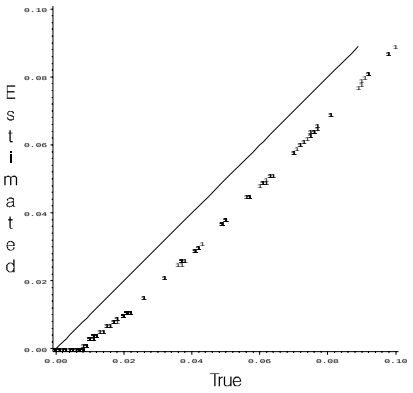

Figure 3. Estimates vs Truth
Cumulative Distribution of Matches
3rd Files, Interaction EM, 1−1

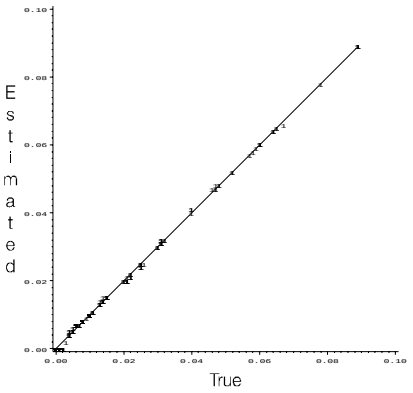

Figure 4. Estimates vs Truth
Cumulative Distribution of Nonmatches
1st Files, Interaction EM, 1−1

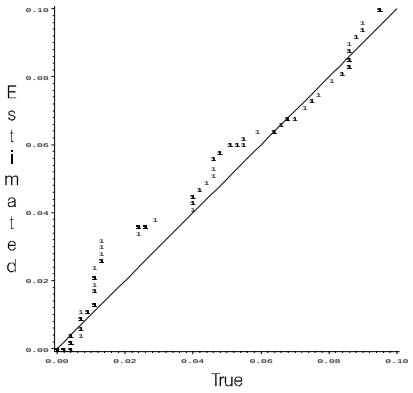

Figure 5. Estimates vs Truth
Cumulative Distribution of Nonmatches
2nd Files, Interaction EM, 1−1

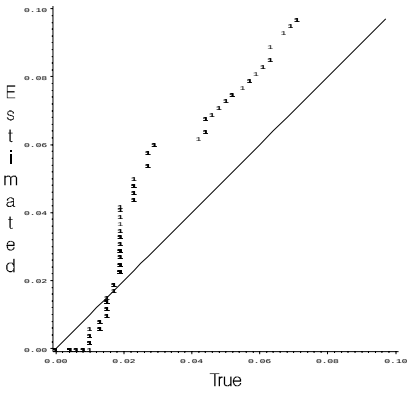

Figure 6. Estimates vs Truth
Cumulative Distribution of Nonmatches
3rd Files, Interaction EM, 1−1

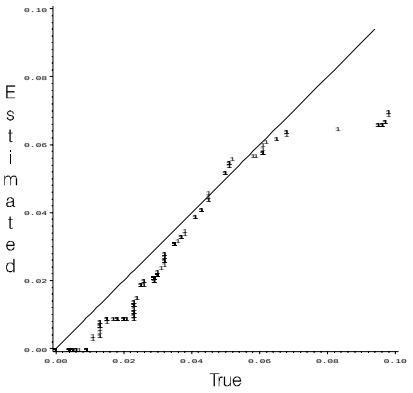

Figure 7. Estimates vs Truth
Cumulative Distribution of Matches
1st Files, Independent EM, 1−1

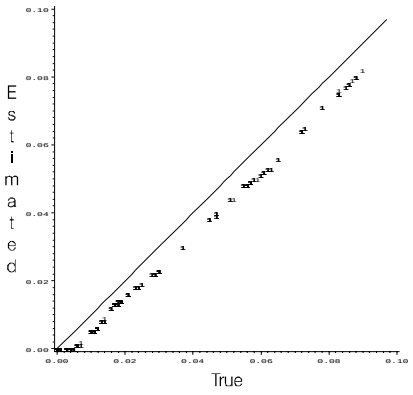

Figure 8. Estimates vs Truth
Cumulative Distribution of Matches
2nd Files, Independent EM, 1−1

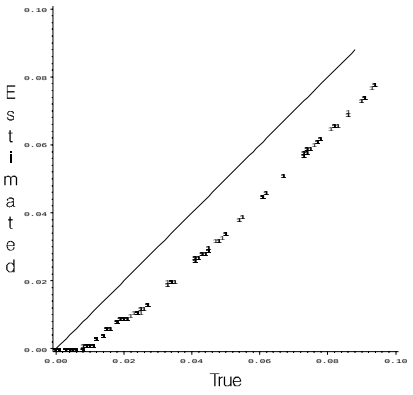

Figure 9. Estimates vs Truth
Cumulative Distribution of Matches
3rd Files, Independent EM, 1−1

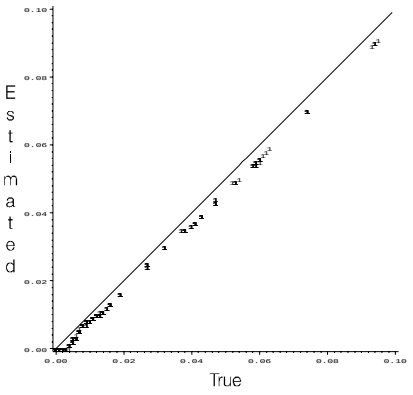

Figure 10. Estimates vs Truth
Cumulative Distribution of Nonmatches
1st Files, Independent EM, 1−1

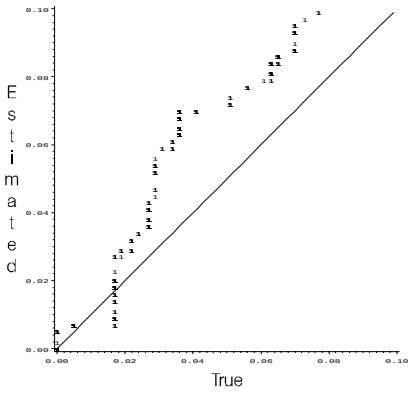

Figure 11. Estimates vs Truth
Cumulative Distribution of Nonmatches
2nd Files, Independent EM, 1−1

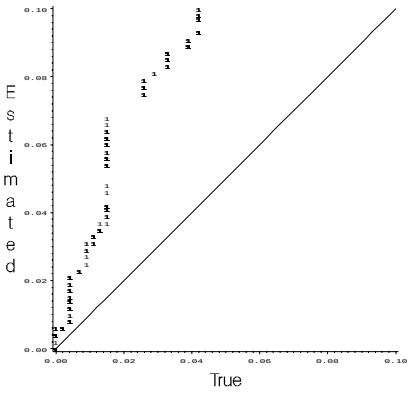

Figure 12. Estimates vs Truth
Cumulative Distribution of Nonmatches
3rd Files, Independent EM, 1−1

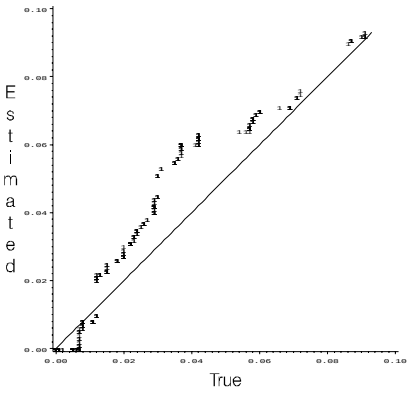

Figure 13. Estimates vs Truth
Cumulative Distribution of Nonmatches
1st Files, Independent EM, 1−1, TB

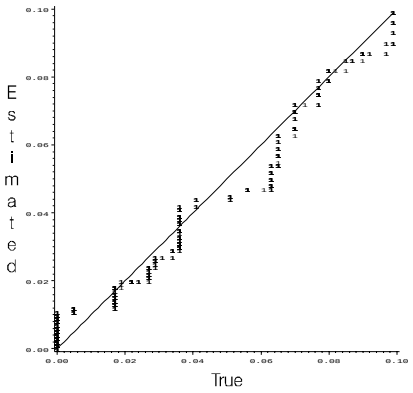

Figure 14. Estimates vs Truth
Cumulative Distribution of Nonmatches
2nd Files, Independent EM, 1−1, TB

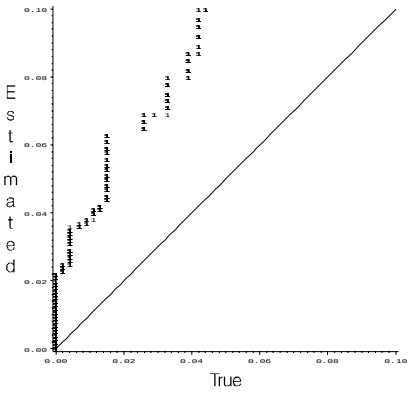

Figure 15. Estimates vs Truth
Cumulative Distribution of Nonmatches
3rd Files, Independent EM, 1−1, TB