RESEARCH REPORT SERIES
(Statistics #2006-2)

Overview of Record Linkage and Current Research Directions

William E. Winkler

Statistical Research Division
U.S. Census Bureau
Washington, DC 20233

Report Issued: February 8, 2006

Overview of Record Linkage and Current Research Directions

William E Winkler[1], 2005Nov15 6p
U.S. Bureau of the Census
Statistical Research, Room 3000-4
Washington, DC 20233-9100
william.e.winkler@census.gov

Abstract

This paper provides background on record linkage methods that can be used in combining data from a variety of sources such as person lists business lists. It also gives some areas of current research.

1. Introduction

Record linkage is the means of combining information from a variety of computerized files. It is also referred to as *data cleaning* (McCallum and Wellner 2003) or *object identification* (Tejada et al. 2002). The basic methods compare name and address information across pairs of files to determine those pairs of records that are associated with the same entity. An entity might be a business, a person, or some other type of unit that is listed. Based on economic relationships, straightforward extensions of methods might create functions and associated metrics for comparing information such as receipts or taxable income. The most sophisticated methods use information from multiple lists (Winkler 1999b), create new functional relationships between variables in two files that can be associated with new metrics for identifying corresponding entities (Scheuren and Winkler 1997), or use graph theoretic ideas for representing linkage relationships as conditional random fields that be partitioned into clusters representing individual entities (McCallum and Wellner 2003, Wei 2004).

The most basic application is identifying duplicates within a file or identifying duplicates across two files. If a single large file is considered, then the record linkage or matching procedures may be intended to identify duplicates. The duplicates can have the effect of erroneously inflating estimates of the number of entities in different categories. For instance, in a list of business, duplicates would inflate estimates in different industrial categories. The duplicates could also cause the number of individuals employed in a set of different firms to be overestimated. If a larger file is being updated using information from a more current but smaller file, then the smaller file is used to obtain records of new entities. The smaller file may contain information about firms or businesses in various categories such as finance or services that may be underrepresented in the larger file. In some situations, a combination of a large amount of duplication and undercoverage may cause severe errors in any uses of the list and the quantitative data that is associated with the list.

If a number of files are combined into a data warehouse, then Fayad and Uthurusamy (1996, 2002) and Fayad et al. (1996) have stated that the majority (possibly above 90%) of the work is associated with cleaning up the duplicates. Winkler (1995) has shown that computerized record linkage procedures can significantly reduce the resources needed for identifying duplicates in comparison with methods that are primarily manual. Newcombe and Smith (1975) have demonstrated the purely computerized duplicate detection in high quality person lists can often

[1] Disclaimer: This report is released to inform interested parties of research and to encourage discussion. The views expressed are those of the author and not necessarily those of the U. S. Census Bureau.

identify duplicates at greater level of accuracy than duplicate detection that involves a combination of computerized procedures and review by highly trained clerks. The reason is that the computerized procedures can make use of overall information from large parts of a list. For instance, the purely computerized procedure can make use of the relative rarity of various names and combinations of information in identifying duplicates. The relative rarity is computed as the files are being matched. Winkler (1995, 1999a) observed that the automated frequency-based (or value-specific) procedures could account for the relative rarity of a name such as 'Martinez' in cities such as Minneapolis, Minnesota in the US in comparison with the relatively high frequency of "Martinez' in Los Angeles, California.

In a very large 1990 Decennial Census application, the computerized procedures were able to reduce the need for clerks and field follow-up from an estimated 3000 individuals over 3 months to 200 individuals over 6 weeks (Winkler 1995). The reason for the need for 200 clerks is that both first name and age were missing from a small proportion of Census forms and Post Enumeration Survey forms. If the clerks cannot supply the missing data from auxiliary information, then a field visit is often needed to get the missing information. The Post Enumeration Survey (PES) provided an independent re-enumeration of a large number of blocks (small Census regions) that corresponded to approximately 70 individuals. The PES was matched to the Census so that a capture-recapture methodology could be used to estimate both undercoverage and overcoverage to improve Census estimates. In the 1992 U.S. Census of Agriculture, the computerized procedures were able to reduce the clerical review from 75 clerks over 3 months to 6500 hours of review (Winkler 1995). The entities in the Census of Agriculture lists are farms corresponding to individuals, various types of partnerships of individuals, and corporations. Based on a large validation follow-up of the matching, the computerized procedures identified more duplicates automatically than the clerks were able to identify in the previous Census of Agriculture. The duplication in the final file was 2% in contrast to 10% in the previous final file from five years earlier.

In some situations, computerized record linkage can help preserve the confidentiality of information in a particular file or in a group of files. The record linkage procedures can delineate records that are at risk of disclosure. To deal with the disclosure risk, Sweeney (1999) has methods for reducing the identifying information in a group of records such as might be available in a hospital or large health network. Individuals with the highest access levels might have nearly full access to information in files. Individuals with lower access levels might automatically have only their access reduced to certain aggregate quantities associated with individual entities. In addition to removing identifiers such as name, social security number, doctor's name, health plan name, all address information, quantities and values associated with certain types of treatments might also be aggregated. Iyengar (2002) provides disclosure-risk-reduction procedures that reduce identifying information in an optimal manner while still preserving specific analytic properties of the files.

Abowd and Woodcock (2002, 2004) have built large files of business entities that are combined with persons for the purpose of studying employment and labor dynamics. To allow use of the files by other economists, they used a combination of record linkage and micro-data confidentiality procedures to identify at risk records and mask data associated with them. Other economists can use the semi-confidential files to develop models and software that are used on the original confidential data. The results of the analyses such as tables and regression coefficients on the confidential data are given an additional review before publication. Abowd and Vilhuber (2005) have observed that the effect of very small amounts of error in identifiers can have small effects of some estimates and relatively large effects on other estimates. For instance, in a file of a billion (10^9) records representing quarterly employment in the State of California for twenty years, the number of erroneous social security numbers (SSNs) is approximately 1-2% per quarter. If the SSNs are not corrected using record linkage methods, then there would be approximately two breaks in every time-series associated with an individual.

The breaks in the time-series can drastically affect the estimates of job creation, job loss, and employment rates that are based on the files.

The outline of this paper is as follows. The second section gives more background on the types of record linkage that are currently being applied. The third section provides details of the record linkage model that was introduced by Newcombe (1959, 1962) and given a formal mathematical foundation by Fellegi and Sunter (1969). The basic ideas are based on statistical concepts such as odds ratios, hypothesis testing, and relative frequency. Because much of the data is based on textual information such as names and addresses, most of the advances in record linkage methods have been in the computer science literature. In particular, methods of string comparison for accounting for typographical error (Winkler 1990a, Cohen et al. 2003a,b), methods of parsing names and addresses into components that correspond and can be more readily compared (e.g., Winkler 1995, Borkar et al. 2001, Christen et al. 2002, Churches et al. 2002), and automated methods of obtained optimal record linkage without training data (Winkler 1989a, 1993b) and with training data (Bilenko et al. 2003). The fourth section covers a number of methods of research for improving matching efficacy. Many of the ideas are being applied in particular settings. The combination of methods is likely to yield significant improvements in matching business lists. Business lists are typically more difficult to match than other types of lists such as person lists or agriculture lists because of the variants of names and the variants of addresses (Winkler 1995). The fifth section is concluding remarks.

2. **Background**

Record linkage of files (Fellegi and Sunter 1969) is used to identify duplicates when unique identifiers are unavailable. It relies primarily on matching of names, addresses, and other fields that are typically not unique identifiers of entities. Matching businesses using business names and other information can be particularly difficult (Winkler 1995). Record linkage is also called object *identification* (Tejada et al. 2001, 2002), *datacleaning* (Do and Rahm 2000), *approximate matching* or *approximate joins* (Gravanao et al. 2001, Guha et al.2004), *fuzzy matching* (Ananthakrisha et al. 2002), and *entity resolution* (Benjelloun et al. 2005).

Rahm and Do (2000) provided an overview of datacleaning and some resesearch problems. Tejada et al. (2001, 2002) showed how to define linkage rules in a database environment. Hernandez and Stolfo (1995) gave merge/purge methods for matching in a database situation. Sarawagi and Bhamidipaty (2002) and Winkler (2002) demonstrated how machine-learning methods could be applied in record linkage situations where training data (possibly small amounts) were available. Ananthakrishna et al. (2002) and Jin et al. (2002, 2003) provided methods for linkage in very large files in the database environment. Cohen and Richman (2002) showed how to cluster and match entity names using methods that are scalable and adaptive. Cohen et al. (2003a,b) provide new methods of adaptive string comparators based on hidden Markov models that improve on the non-adaptive string comparators in a number of situations. Bilenko and Mooney (2003) provide adaptive, Hidden Markov methods that both standardize and parse names and addresses while comparing and matching components of them. Borkar et al. (2001), Christen et al. (2002), and Churches et al. (2002) use Hidden Markov models for adaptive name and address standardization. Wei (2004) provides new Markov edit algorithms that should improve over the algorithms that apply basic Hidden Markov models for string comparison. Lafferty et al. (2001), McCallum and Wellner (2003), and Culotta and Mcallum (2005) used conditional random fields for representing an exceptionally large number of linkage relationships in which the likelihoods are optimized via Markov Chain Monte Carlo (MCMC) and graph partitioning methods. Ishikawa (2003) provides on a more general characterization of Markov random fields, graph partitioning, and optimization of likelihoods that may yield faster computational algorithms. Chaudhuri et al. (2005) provide a theoretic characterization of

matching in terms of generalized distances and certain aggregates. Benjelloun et al. (2005) provide a characterization of how pairs are brought together during a matching process.

In a substantial number of situations, the files are too big to consider every pair in the cross product space of all pairs from two files. Newcombe (1962, 1988) showed how to reduce the number of pairs considered by only considering pairs that agreed on a characteristic such as surname or date-of-birth. Such reduction in the number of pairs is called *blocking*. Hernandez and Stolfo (1995) also showed how to use multiple passes on a database to bring together pairs. Each pass corresponds to a different sort ordering of the files and pairs are only considered in a sliding window of fixed size. After the pairs are brought together more advanced, compute-intensive methods are used for comparing them. McCallum et al. (2000) showed that the first step should be a clustering step that is performed with an easily computable, fast method (referred to as *canopies*) and the second step can use more expensive computational methods. Chaudhuri et al (2003) showed how to create index structures that allow for certain types of typographical error in matching within a database. In their application, their methods reduced computation by a factor of three in comparison with naïve methods that compare every pair of records in two files. Baxter et al. (2003) used a more easily applied method based on *q*-grams (described later). Still more advanced methods rely of embedding the files and associated string comparators for approximate string comparison in versions of *d*-dimensional Euclidean space and using sophisticated *R*-tree bi-comparison searches (Jin et al. 2003). With the possibility of significant computation associated with the embedding algorithms, the methods are intended to reduce computation associated with comparing pairs from $O(N^2)$ to $O(N)$ or $O(N\log N)$ where N is the size of the file being searched. Yancey and Winkler (2004) developed BigMatch technology for matching moderate size lists of 100 million records against large administrative files having upwards of 4 billion records. The methods are faster than the other methods because they rely on models of typographical error corresponding to actual names and addresses that are not always explicitly used in some of the other new methods.

We illustrate some of the issues of record linkage with a straightforward example. The following three pairs represent three individuals. In the first two cases, a human being could generally determine that the pairs are the same. In both situations, the individuals have reasonably similar names, addresses, and ages. We would like software that automates the determination of match status. In the third situation, we may know that the first record of the pair was a medical student at the university twenty years ago. The second record is from a current list of physicians in Detroit who are known to have attended the University of Michigan. It is associated with a doctor in a different city who is known to have attended medical school at the

Table 1. Elementary Examples of Matching Pairs Of Records (Dependent on Context)

Name	Address	Age
John A Smith	16 Main Street	16
J H Smith	16 Main St	17
Javier Martinez	49 E Applecross Road	33
Haveir Marteenez	49 Aplecross Raod	36
Gillian Jones	645 Reading Aev	24
Jilliam Brown	123 Norcross Blvd	43

university. With good automatic methods, we could determine that the first two pairs represent the same person. With a combination of automatic methods and human understanding, we might determine the third pair is the same person.

The following example describes a situation of survey frame deficiencies that might occur in a list of small businesses or of farms. The two components of a survey frame are the list of entities to be surveyed and the quantitative and other data associated with the list that may be used for sampling. The list may consist of names and addresses with additional information such as contact person and source of the name and address. It is quite possible that there is undercoverage and duplication in the list. For instance, if the list of businesses consists of entities selling petroleum products, then there may be turnover of 20% per year in the names and addresses. In some situations, a small company such as a gasoline station (retailer) may have gone out of business. In many other situations, the company may have changed its address to that of its accountant or changed from a location address to an owner address. The owner and accountant addresses can change from year to year. In a set of situations, the name and address associated with the entity may have changed. In the situation of a company going out of business, a source of entities doing the same type of business such as a gasoline station is needed. If the survey frame is not updated with new entities, then it is possible that the list of companies miss upwards of 30-40% of the universe of gasoline stations if the turnover rate is 20% per year. If a survey is intended to collect total sales of different products and uses the list that is missing a sizeable portion of the universe, then it will yield survey estimates that are biased much lower than the truth.

A second set of situations is illustrated in Table 2. The first name refers to the business name and its physical location. The second is the name of the owner of the business with his home address. The third is the address of the accountant who does the books for the company. The name 'J A S, Inc' is an abbreviation of the actual name of the business 'John A Smith, Inc' that owns the gasoline station. It is possible that different lists associated with the set of businesses may have entries corresponding to anyone of the listed forms of the entity that is the gasoline station. In situations where different source lists are used in updating the survey frame that all three addresses may be on the frame. In some situations, all three forms of the address may be surveyed. Some duplication may be corrected when individuals return the survey form as being a

Table 2 Examples of Names and Addresses Referring to the Same Business Entity

Correspondence Address	Description
Main Street Philips Service 1623 Main Street Anytown, OH	Physical location of business
John A Smith 761 Maple Drive SuburbTown1, OH	Owner of small business, lives in suburb on Anytown, OH
J A S, Inc c/o Robert Jones, CPA 1434 Lee Highway, Suite 16 SuburbTown2, OH	Incorporated name of business, accountant does business' books and government forms

duplicate. In the case of a sample survey, very little of the duplication can be corrected through the survey operation (e.g., Winkler 1995). An organization such as a federal government may have the resources to maintain a complete survey frame or population file with name and addresses variants with appropriate dates associate with the variants. In that type of situation, the large population file might be used in correcting smaller lists.

The second component of the survey frame is the quantitative and other information associated with the lists of entities. For instance, total sales might be the quantitative field associated with a list of gasoline stations. If a sizeable portion (say 5%) of the quantities is in error by a factor of ten (either low or high), then it would be very difficult to use the list and quantities as a sampling frame. The total sales in the frame would be used in creating strata and associated sampling weights. If the total sales returned on the survey form were correct, then any estimates could be severely in error. We can derive two fundamental points from the example. If all of the quantitative data associated with a survey frame is correct and the survey frame has substantial undercoverage and overcoverage, then the errors in any estimates due to frame errors can exceed all other sources combined. If the frame is correct and 5% of the quantitative items are in error (possibly substantially), then the errors in estimates due to the deviation of the quantitative fields in the survey frame may exceed all other sources of error such as sampling. Winkler (2003a) presents an overview of edit/imputation methods that might be used in correcting discrete data from demographic surveys. Bruni (2004, 2005) provides methods for dividing continuous data into discrete ranges into which discrete-data methods can be applied. We can use relatively clean quantitative data associated with a group of entities to clean-up the duplication and coverage in the basic list. We note that the issues of undercoverage, overcoverage, and errors in the information associated with a list can occur with an administrative list or a group of files that interact or co-operate in a manner to allow analyses.

3. **Record linkage**

In the section, we provide background on the Fellegi-Sunter model of record linkage, name and address standardization, string comparators for approximate string comparison, methods of parameter estimation, and search and retrieval mechanisms.

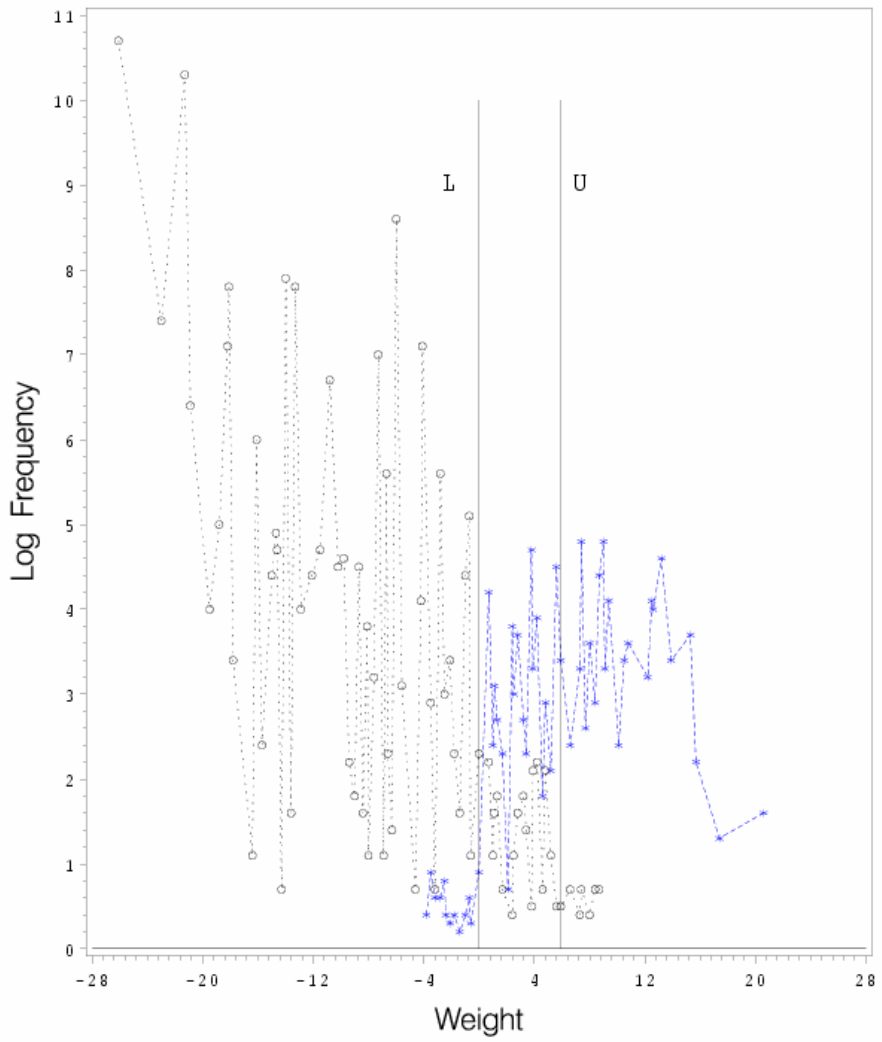
3.1 **The Fellegi-Sunter model of record linkage**

Fellegi and Sunter (1969) provided a formal mathematical model for ideas that had been introduced by Newcombe (1959, 1962, see also 1988). They provided many ways of estimating key parameters. The methods have been rediscovered in the computer science literature (Cooper and Maron 1978) but without proofs of optimality. To begin, notation is needed. Two files **A** and **B** are matched. The idea is 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, making rigorous concepts introduced by Newcombe (1959), considered ratios of probabilities of the form:

$$R = P(\gamma \in \Gamma \mid M) / P(\gamma \in \Gamma \mid U) \qquad (1)$$

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 ratio R or any monotonely increasing function of it such as the natural log is referred to as a matching weight (or score).

Figure 1. Plot of Weight versus Log Frequency for Nonmatches and Matches



The decision rule is given by:

If $R > T_{\mu}$, then designate pair as a match.

If $T_\lambda \le R \le T_\mu$, then designate pair as a possible match
and hold for clerical review. (2)

If $R < T_\lambda$, then designate pair as a nonmatch.

The cutoff thresholds T_μ and T_λ are determined by a priori error bounds on false matches and false nonmatches. Rule (2) agrees 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 (1) would be large. On the other hand, if $\gamma \in \Gamma$ consists primarily of disagreements, then ratio (1) would be small. Rule (2) partitions the set $\gamma \in \Gamma$ into three disjoint subregions. The region $T_\lambda \le R \le T_\mu$ is referred to as the no-decision region or clerical review region. In some situations, resources are available to review pairs clerically. Figure 1 provides an illustration of the curves of log frequency versus log weight for matches and nonmatches, respectively. The two vertical lines represent the lower and upper cutoffs thresholds T_λ and T_μ, respectively.

3.2 **Name and address standardization**

Standardization consists of replacing various spelling of words with a single spelling. For instance, different spellings and abbreviations of 'Incorporated' might be replaced with the single standardized spelling 'Inc.' The standardization component of software might separate a general string such as a complete name or address into words (i.e., sets of characters that are separated by spaces and other delimiters). Each word is then compared lookup tables to get standard spelling. The first half of the following table shows various commonly occurring words that are replaced by standardized spellings (given in capital letters). After standardization, the name string is parsed into components (second half of the following table) that can be compared. The examples are produced by general name standardization software (Winkler 1993a) for the US Census of Agriculture matching system. Because the software does well with business lists and person matching, it has been used for other matching applications at the Census Bureau and other agencies. At present, it is not clear that there is any commercial software for name standardization. Promising new methods based on Hidden Markov models (Borkar et al. 2001, Churches et al. 2002, Christen et al. 2002) may improve over the rule-based name standardization in Winkler (1993a). Although the methods clearly improve over more conventional address standardization methods for difficult situations such as Asian or Indian addresses, they did not perform as well as more conventional methods of name standardization.

Table 3. Examples of Name Parsing

	Standardized
1.	DR John J Smith MD
2.	Smith DRY FRM
3.	Smith & Son ENTP

	Parsed						
	PRE	FIRST	MID	LAST	POST1	POST2	BUS1	BUS2
1.	DR	John	J	Smith	MD			
2.				Smith			DRY	FRM
3.				Smith		Son	ENTP	

The following table illustrates address standardization with a proprietary package developed by the Geography Division at the U. S. Census Bureau. In testing in 1994, the software significantly outperformed the best U. S. commercial packages in terms of standardization rates while producing comparably accurate standardizations. The first half of the table shows a few addresses that have been standardized. In standardization, commonly occurring words such as 'Street' are replaced by an appropriate abbreviation such as 'St' that can be considered a standard spelling that may account for some spelling errors. The second half of the table represents components of addresses produced by the parsing. The general software produces approximately fifty components. The general name and address standardization software that we make available with the matching software only outputs the most important components of the addresses.

Table 4. Examples of Address Parsing

	Standardized
1.	16 W Main ST APT 16
2.	RR 2 BX 215
3.	Fuller BLDG SUITE 405
4.	14588 HWY 16 W

	Parsed									
	Pre2	Hsnm	Stnm	RR	Box	Post1	Post2	Unit1	Unit2	Bldg
1.	W	16	Main			ST		16		
2.				2	215					
3.									405	Fuller
4.		14588	HWY 16				W			

3.3 **String comparators**

In many matching situations, it is not possible to compare two strings exactly (character-by-character) because of typographical error. Dealing with typographical error via approximate string comparison has been a major research project in computer science (see e.g., Hall and Dowling 1980, Navarro 2001). In record linkage, one needs to have a function that represents approximate agreement, with agreement being represented by 1 and degrees of partial agreement being represented by numbers between 0 and 1. One also needs to adjust the likelihood ratios (3) according to the partial agreement values. Having such methods is crucial to matching. For instance, in a major census application for measuring undercount, more than 25% of matches would not have been found via exact character-by-character matching. Three geographic regions are considered in Table 5. The function Φ represents exact agreement when it takes value one and represents partial agreement when it takes values less than one. In the St Louis region, for instance, 25% of first names and 15% of last names did not agree character-by-character among pairs that are matches.

Jaro (1989) introduced a string comparator that accounts for insertions, deletions, and transpositions. The basic Jaro algorithm has three components: (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 half the length of the shorter string. The definition of transposition is that the character from one string is out of order with the corresponding common character from the other string. The string comparator value (rescaled for consistency with the practice in computer science) is:

$$\Phi_j(s_1, s_2) = 1/3(N_C/len_{s1} + N_C/len_{s2} + 0.5N_t/N_C),$$

where s_1 and s_2 are the strings with lengths len_{s1} and len_{s2}, respectively, N_C is the number of common characters between strings s_1 and s_2 where the distance for common is half of the minimum length of s1 and s_2, and N_t is the number of transpositions. The number of transpositions N_t is computed somewhat differently from the obvious manner.

Table 5. Proportional Agreement
By String Comparator Values
Among Matches
Key Fields by Geography

	StL	Col	Wash
First			
$\Phi = 1.0$	0.75	0.82	0.75
$\Phi \geq 0.6$	0.93	0.94	0.93
Last			
$\Phi = 1.0$	0.85	0.88	0.86
$\Phi \geq 0.6$	0.95	0.96	0.96

Using truth data sets, Winkler (1990a) introduced methods for modeling how the different values of the string comparator affect the likelihood (1) in the Fellegi-Sunter decision rule. Winkler (1990a) also showed how a variant of the Jaro string comparator Φ dramatically improves matching efficacy in comparison to situations when string comparators are not used. The variant employs some ideas of Pollock and Zamora (1984) in a large study for the Chemical Abstracts Service. They provided empirical evidence that quantified how the probability of keypunch errors increased as the character position in a string moved from the left to the right.

More recent work by Sarawagi and Bhamidipaty (2002) and Cohen et al. (2003a,b) provides empirical evidence that the new string comparators can perform favorably in comparison to Bigrams and Edit Distance. Edit Distance uses dynamic programming to determine the minimum number of insertions, deletions, and substitutions to get from one string to another. The Bigram metric counts the number of consecutive pairs of characters that agree between two strings. A generalization of bigrams is *q*-grams where *q* can be greater than 3. The recent hybrid string comparator of Cohen et al. (2003b) uses variants of the TFIDF metrics from the Information Retrieval literature. The basic TFIDF makes uses of the frequency of terms in the entire collections of records and the inverse frequency of a specific term in a record. The metric TFIDF has some relationship to value-specific or frequency-based matching of Newcombe (1959, 1962). Cohen et al. (2003a,b) generalize TFIDF to soft TFIDF with a heuristic that partially accounts for certain kinds of typographical error. Alternate methods of accounting for relative frequency that also account for certain kinds of typographical error are due to Fellegi and Sunter (1969), Winkler (1989b), and Chaudhuri et al. (2003). The soft TFIDF metric of Cohen et al. (2003a,b) will likely outperform the Jaro-Winkler comparator in some types of lists in terms of distinguishing power while being slower to compute. Yancey (2003, 2005), however, has demonstrated that the newer

string comparators do not outperform the original Jaro-Winkler string comparator on typical large Census applications. McCallum et al. (2005) apply conditional random fields to the problem of modeling string comparator distances.

Table 6 compares the values of the Jaro, Winkler, Bigram, and Edit-Distance values for selected first names and last names. Bigram and Edit Distance are normalized to be between 0 and 1. All string comparators take value 1 when the strings agree character-by-character. The renormalization is consistent with the approach of Bertolazzi et al. (2003) or Cohen et al. (2003a,b).

Table 6. Comparison of String Comparators Using Last Names and First Names

Two strings		String comparator Values			
		Jaro	Winkler	Bigram	Edit
SHACKLEFORD	SHACKELFORD	0.970	0.982	0.925	0.818
DUNNINGHAM	CUNNIGHAM	0.896	0.896	0.917	0.889
NICHLESON	NICHULSON	0.926	0.956	0.906	0.889
JONES	JOHNSON	0.790	0.832	0.000	0.667
MASSEY	MASSIE	0.889	0.933	0.845	0.667
ABROMS	ABRAMS	0.889	0.922	0.906	0.833
HARDIN	MARTINEZ	0.000	0.000	0.000	0.143
ITMAN	SMITH	0.000	0.000	0.000	0.000
JERALDINE	GERALDINE	0.926	0.926	0.972	0.889
MARHTA	MARTHA	0.944	0.961	0.845	0.667
MICHELLE	MICHAEL	0.869	0.921	0.845	0.625
JULIES	JULIUS	0.889	0.933	0.906	0.833
TANYA	TONYA	0.867	0.880	0.883	0.800
DWAYNE	DUANE	0.822	0.840	0.000	0.500
SEAN	SUSAN	0.783	0.805	0.800	0.400
JON	JOHN	0.917	0.933	0.847	0.750
JON	JAN	0.000	0.000	0.000	0.667

3.4 **Heuristic improvement by forcing 1-1 matching**

In a number of situations, matching can be improved by forcing 1-1 matching. In 1-1 matching, a record in one file can be matched with at most one record in another file. Some early matching systems applied a greedy algorithm 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. Jaro (1989) provided a *linear sum assignment procedure* (lsap) to force 1-1 matching because he observed that greedy algorithms often made a greater number of erroneous assignments. In the following (Table 7), the two households are assumed to be the same, individuals have substantial identifying information, and the ordering is as shown. An lsap algorithm causes the wife-wife, son-son, and daughter-daughter assignments correctly because it optimizes the set of assignments globally over the household. Other algorithms such as greedy algorithms can make erroneous assignments such as husband-wife, wife-daughter, and daughter-son.

Table 7. Representation of
A Household

HouseH1	HouseH2
husband	
wife	wife
daughter	daughter
son	son

Table 8 illustrates the assignment procedure using matrix notation. c_{ij} is the (total agreement) weight from matching the i^{th} person from the first file with the j^{th} person in the second file. Winkler (1994) introduced a modified assignment algorithm that uses 1/500 as much storage as the original algorithm and is of equivalent speed. The modified assignment algorithm does not induce a very small proportion of matching error (0.1-0.2%) that is caused by the original assignment algorithm. The modified algorithm is useful because many applications consist of situations where a small list is matched against a much larger list. In the situation where one list consists of a set of US Postal ZIP codes containing 50-60,000 entries in each ZIP code, the conventional lsap needed 40-1000 Megabytes of memory that was often not available on smaller machines.

Table 8. Weights (Matching Scores) Associated
With Inidividuals Across Two Households

c_{11} c_{12} c_{13}	4 rows, 3 columns
c_{21} c_{22} c_{23}	Take at most one in each
c_{31} c_{32} c_{33}	row and column
c_{41} c_{42} c_{43}	

3.5 **Automatic and semi-automatic parameter and error-rate estimation**

Fellegi and Sunter (1969) introduced methods for estimating optimal parameters (probabilities) in the likelihood ratio (1). They observed that

$$P(\gamma) = P(\gamma \mid M)\, P(M) + P(\gamma \mid U)\, P(U) \qquad (3)$$

where $\gamma \in \Gamma$ is an arbitrary agreement pattern and M and U are two classes of matches and nonmatches. If the agreement pattern $\gamma \in \Gamma$ is from three fields that satisfy a conditional independence assumption, then the system of seven equations and seven unknowns can be used to estimate the m-probabilities $P(\gamma \mid M)$, the u-probabilities $P(\gamma \mid U)$, and the proportion $P(M)$. The conditional independence assumption corresponds exactly to the naïve Bayes assumption in machine learning (Winkler 2000, Mitchell 1997). Winkler (1988) showed how to estimate the probabilities using the EM-Algorithm (Dempster et al. 1977). Although this is a method of unsupervised learning (e.g., Mitchell 1997, Winkler 1993b) that will not generally find two classes C_1 and C_2 that correspond to M and U, Winkler (1989a) demonstrated that a properly applied EM algorithm provides suitable estimates of optimal parameters in a number of situations. The best situations are when the observed proportion of matches $P(\gamma)$ are computed

over suitably chosen sets of pairs that includes the matches, clerical pairs and upper portion of the nonmatches given in decision rule (2). Because the EM algorithm is an unsupervised learning method for latent classes, the proportion of matched pairs in the set of pairs to which the EM is applied should be above 5%. Recent extensions of the methods for choosing the pairs above a cutoff weight with initial guesses at the parameters are due to Yancey (2002) and Efelkey et al. (2002). The problem is particularly difficult because the optimal m- and u-probabilities can vary substantially from one region of the U.S. to another (Winkler (1989a). In particular, the conditional probability P(agreement on first name | M) can differ significantly from an urban region to an adjacent suburban region. A large portion of the variation is due to the different typographical error rates in different files.

Belin and Rubin (1995) introduced methods of automatic-error rate estimation that used information from the basic matching situations of Winkler (1989a). Their error rate estimates were sufficiently accurate with certain types of high quality population files. Scheuren and Winkler (1993) could use the estimated error rates in a statistical model that adjusts regression analyses for linkage errors. Lahiri and Larsen (2005) extended the Scheuren-Winkler model with a more complete theoretical development of the bias-adjustment procedures. After attempting to apply the Belin-Rubin methods to business files, agriculture files, and certain types of poor quality person files, Winkler (1999a) observed, however, that the Belin-Rubin methods only worked well in a narrow range of situations where the curves associated with matches M and nonmatches U were well-separated and had several other desirable properties. Using simulation with artificial data, Belin and Rubin (1995) had observed a similar lack of applicability in some situations.

Extensions of the basic parameter estimation (unsupervised learning) have been to situations where the different fields used in the EM-algorithm can have dependencies upon one another and when various convex constraints force the parameters into subregions of the parameter space (Winkler 1993b, 1990b). The general fitting algorithm of Winkler (1990b) generalizes the iterative scaling algorithm of Della Pietra et al. (1997). Recent unsupervised learning methods of Ravikumar and Cohen (2004) improve on the methods of Winkler with certain kinds of files and are even competitive with supervised learning methods in some situations. Wang et al. (2003) apply maximum entropy models in contrast to the maximum likelihood models to learn the mixtures of distributions. Although maximum entropy yields the same set of local maxima as maximum likelihood, the global maximum from entropy can differ. Wang et al. (2003) claim that the parameters produced by the global entropy procedure can outperform maximum likelihood-estimated parameters in many situations.

Additional extensions are where small amounts of unlabelled data are combined with the unlabelled data used in the original algorithms (Winkler 2000, 2002, Larsen and Rubin 2001, Elfekey et al. 2002). The general methods (Winkler 2002) can be used for data mining to determine what interaction patterns and variants of string comparators and other metrics affect the decision rules. Winkler (2002) suggested the use of very small amounts of training data that are obtained from a sample of pairs representing the clerical review region of equation (3). This small amount of training data is needed because the amount of typographical error in fields varies across different pairs of files. The amount of typographical error cannot be predicted with unlabeled data and significantly affects optimal estimates of matching parameters. The variant that uses unlabelled data with small amounts of labeled data can yield semi-automatic estimates of classification error rates (Winkler 2002). The general problem of error rate estimation is very difficult. It is known as the regression problem (Vapnik 2000, Hastie et al. 2001).

Winkler and Yancey (2006) provide a method for estimating false match rates that does not require training data and is closely related to the semi-supervised training methods of Winkler (2000, 2002). In the application of semi-supervised learning to record linkage (Winkler 2002), a small amount (less than 0.5%) of training data from pairs in the clerical review region of decision rule (2) is combined with the unlabeled data of the ordinary EM procedure (Winkler 1988, 1993).

Figure 2a. Estimates vs Truth, File A
Cumulative Matches, Tail of Distribution
Independent EM, Lambda = 0.2

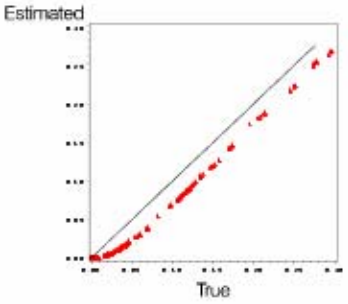


Figure 2b. Estimates vs Truth, File A
Cumulative Nonmatches, Tail of Distribution
Independent EM, Lambda = 0.2

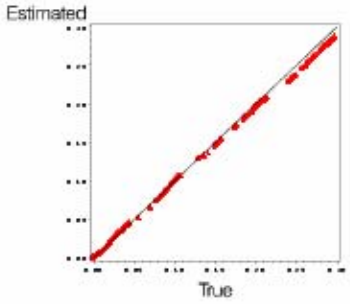


Figure 2c. Estimates vs Truth, File B
Cumulative Matches, Tail of Distribution
Independent EM, Lambda = 0.2

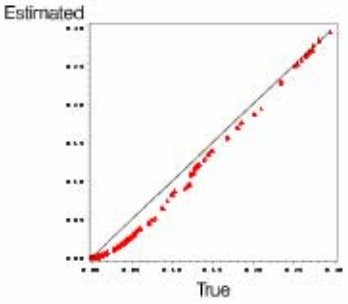


Figure 2d. Estimates vs Truth, File B
Cumulative Nonmatches, Tail of Distribution
Independent EM, Lambda = 0.2

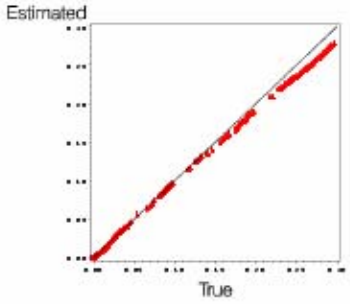


Figure 2e. Estimates vs Truth, File C
Cumulative Matches, Tail of Distribution
Independent EM, Lambda = 0.2

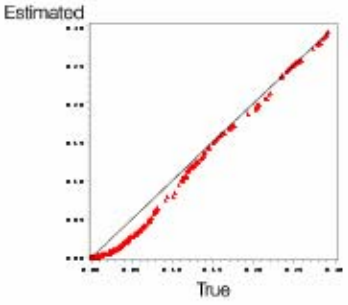
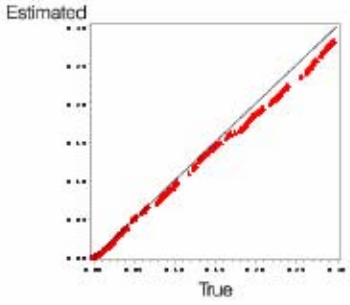


Figure 2f. Estimates vs Truth, File C
Cumulative Nonmatches, Tail of Distribution
Independent EM, Lambda = 0.2



In the newer procedure, all pairs above a certain point of the matching score (or weight) given by equation (1) are designated as a pseudo-true match set of pairs and all pairs below a certain matching score are designated as a pseudo-true nonmatch set of pairs. The unlabelled set of pairs and the 'labeled' set of pairs are then combined in a manner analogous to Winkler (2002). The

plots in Figure 2 illustrate the situation with three test sets for which true match status is known. The cumulative bottom 10% tails of the estimated matches and the cumulative upper 10% tails estimated nonmatches are plotted against the truth that is represented by the 45-degree line. The correspondence between the estimated curves and the truth is better than Winkler (2002) and far better than Belin and Rubin (1995). As Winkler and Yancey (2006) note, the false-match rate estimation procedure is likely only to work effectively in situations where name, address, and other information in two files can effectively be used in bringing together pairs and designating matches accurately. In a number of business-list situations, many (possibly 40-60%) of truly matching pairs have both names and address that differ significantly. In the business-list situations, it is unlikely that the new error-rate estimation procedure will provide suitably accurate estimates.

3.6 **Advanced search and retrieval mechanisms**

In this section, we consider the situation where we must match a moderate size file A of 100 million records with a large file having upwards of 4 billion records. An example of large files might be a Social Security Administrative Numident file having 600 million records, a U.S. Decennial Census file having 300 million records, or a California quarterly employment file for 20 years that contains 1 billion records. If the California employment data has 2-3% percent typographical error in the Social Security Number (SSN) in each quarter, then it is possible that most individuals have two breaks in their 20-year time series. The main ways of correcting the SSNs are by using a combination of name, date-of-birth, and address information. The primary time-independent information is name and date-of-birth because address varies considerably over time. Name variations such as maiden names are sometimes available in the main files or in auxiliary files. Name, date-of-birth, and address can also contain significant typographical error. If first name has 10% typographical error rate and last name, day-of-birth, month-of-birth, andyear-of-birth have 5% typographical error rates, then exact character-by-character matching across quarters could miss 25% of matches.

With classical matching, 10 blocking passes might be performed on the pair of files. For each matching pass, the two files are sorted according to the blocking criteria and then passed through the matching software. To complete the matching, 20 passes would need to be made on each file. BigMatch technology alleviates the limitations of the classical matching situation (Yancey and Winkler 2004, Yancey 2004). Only the smaller B file and appropriate indexes are held in memory. In addition to a copy of the B-file, two sets of indexes are created for each set of blocking criteria. The first index corresponds to the quick sort method of Bentley and Sedgewick (1996). In testing, we found the Bentley and Sedgewick sort to be slightly faster than three other quick sort algorithms. The second index gives a very fast method of retrieving and comparing the records information from the B-file with individual records from the A-file. A B-file of 33 million records with associated sets of indexes can reside in 4 gigabytes of memory. Only one pass is made on the B-file and on the A-file. The pass on the A-file is input/output pass only. The possibly very large A-file is never sorted. Several output streams are created for each set (or group) of blocking criteria. Each individual B-record is compared to all of the appropriate A-records according to the set of blocking criteria. No pair is compared more than once. If the B-file contains 1 billion records, then the BigMatch technology may need only 4 terabytes of disk storage in contrast with 16 or more terabytes using conventional matching. Although the BigMatch software effectively makes 10-blocking passes simultaneously, it is nearly as fast as a classical matching program that only makes a single pass against a pair of files. It processes approximately 100,000 pairs per second. It saves the cpu-time of multiple sorts of the large file that may contain a billion or more records. A single sort of a billion-record file on a fast machine may take 12+ hours. The biggest savings is often from the reduction in the amount of skilled

intervention by programmers who must track a large number of files, make multiple runs, and put together information across multiple runs.

There are several important issues with matching very large files. The first is whether a procedure that brings together all pairs is superior to a procedure in BigMatch that only brings together a much smaller subset of pairs corresponding to a set of blocking criteria. Guha et al. (2004) and Koudas et al. (2004) have procedures that provably bring together all pairs in a pair of files. Both Benjelloun et al. (2005) and Chaudhuri et al. (2003) have alternative procedures for bringing together pairs in a relatively efficient manner. Winkler (2004) examined the issue with 2000 Decennial Census containing 300 million records that also have true matching status for a large subset of regions containing approximately 600,000 matched pairs. Winkler considered the 10^{17} pairs (300 million × 300 million) and demonstrates that 99.5% of the true matches can be located in a subset of 10^{12} pairs determined by 11 blocking criteria. Some of the most difficult missed matches were children in a household headed by a single or separated mother (Table 9). The information associated with the individuals was represented in significantly different form: the children were listed under two different last names, date-of-birth was missing in one representation of the household, and the street address was missing in the other representation of the household. It is interesting to observe the high rate of typographical error that may, at least partially, be due to scanning error. The matching children have no 3-grams in common. Two records have a 3-gram in common if any three consecutive characters from one record can be located in another record. It is unlikely that these most difficult-to-match record pairs could be identified through any computerized procedure that uses only the information in the Census files. Duplicate pairs similar to those represented in Table 9 are typically found during a field follow-up.

Table 9. Example of missed matches (artificial data)

	Household 1		Household 2	
	First	Last	First	Last
HeadH	Julia	Smoth	Julia	Smith
Child1	Jerome	Jones	Gerone	Smlth
Child2	Shyline	Jones	Shayleene	Smith
Child3	Chrstal	Jcnes	Magret	Smith

4. **Current research**

This section considers thirteen areas of current research. The first area consists of methods for automatically or semi-automatically estimating error rates. The second area consists of methods for using information from additional files to improve the matching in two files A and B. The third area covers mechanisms for bringing together pairs in very large files in the presence of moderate or significant typographical error in individual fields. The fourth area is methods for constructing functions $y = f(x)$ where x in A and y in B and associated comparison metrics so that that additional information can be used in improving matching. The fifth area provides methods of creating large linkage graphs of information from multiple files in which the linkage information and the information in fields are compared. In the sixth area, we consider methods of analysis of merged data that partially compensate for linkage error. The seventh area considers the effects of relative frequency of weakly identifying strings, methods for estimating typographical error rates, and situations where lists have low overlap rates. In the eighth area, we consider where additional fields, lack of independence, and typographical error rates can affect matching. The ninth area covers existing classification algorithms that are used for record

linkage. The tenth area provides an overview of methods of string comparison and closely related extensions for longer fields. In the eleventh area, we consider methods of standardization and data extraction. The twelfth area provides methods for maintaining population registers and other large files. In the thirteenth area, we consider how multiple addresses (and other fields tracked over time) can significantly improve matching.

4.1 **Automatic estimation of error rates**

The first step in estimating the quality of linked files A and B is the estimation of rates of false matches and false nonmatches. Winkler (1989a) observed that the optimal parameters $P(A_{fl} \mid M)$ where A_{fl} is agreement or disagreement of field f_l can vary significantly from one pair of files to another and can greatly affect the shape and separation of the aggregate weighting curves in Figure 1 (section 3.1). This is true even if the two pairs of files have the same matching fields and represent adjacent urban and suburban regions. The probability $P(A_{fl} \mid M)$ depends on typographical error rates that are often difficult to determine without training data. In a few easier situations, Fellegi and Sunter (1969) and Winkler (1989b) have provided some intuition on how to get crude estimates of typographical error rates.

In obtaining estimates of error rates, Belin and Rubin (1995) assumed that the curves associated with matches and nonmatches were each unimodal and quite well separated. Unlike some high-quality population files in which the curves separate, Winkler (1999a) observed that the curves associated with pairs of business and agriculture lists overlap significantly and are not unimodal. If there is name or address standardization failure, the curves associated with matches can be multi-modal. This situation of multi-modality can occur if one of the source lists is created from several different lists. It can even occur if different keypunchers have varying skills or learn to override some of the edits in the keypunch software. The basic research problem is characterizing under which conditions errors can be estimated automatically. Alternatively, Larsen and Rubin (2001) and Winkler (2002) haves shown that error rates can be estimated if combinations of labeled training data and unlabelled data are used in the estimation. The labeled training data can be a very small proportion of the pairs for which the true match status is known. Typically, the sampled set of labeled pairs is concentrated in the clerical review region given by classification rule (2). In many situations it may be infeasible to obtain a subset of the pairs for training data.

If there are no training data, then Winkler and Yancey (2006) have demonstrated that accurate estimates of error rates can be achieved in a narrow range of situations where non-1-1-matching methods are used. The situations primarily include those where two curves are reasonably separated. One curve is based on a reasonably pure subset of matches that separates from a curve that is based on the the remaining pairs. By labeling the pairs above a given matching score as matches and pairs below a given score as nonmatches, pseudo-training data can be obtained that can be combined with all of the remaining data (which is unlabelled) in a procedure that mimics semi-supervised learning. One reason that this procedure does not work in general is that it is not always possible to easily separate a large subset of pairs that almost exclusively contains matches. This non-separate-curve situation can occur with many files of individuals and typically occurs with files of businesses or agricultural entities. It occurs because of different ways of representing the names and addresses. A research problem is whether it is possible to having a scoring mechanism for matching that yields (somewhat) accurate estimates of false match rates?

Larsen (2005) has developed theory for estimating error rates in 1-1 matching situations. Accuracy of 1-1 matching is often better than in non-1-1-matching situations (Winkler 1994). Matching with 1-1 methods often will quickly obtain many matches with high matching scores and get other residual matches with lower scores automatically. It is known to improve in comparison with non-1-1 matching. If the 1-1 matching is forced via certain types of linear sum

assignment procedures that are applied to the weights of pairs, then the restraint situation is very complicated. The matching weights need to vary according to a very complicated restraint structure rather than the simplistic restraint structure of non-1-1 matching. In an appropriately parameterized 1-1 matching situation (Larsen 2005), the weight of each pair depends on the weights of a large number of other pairs. Larsen proposes a non-trivial Metropolis-Hastings procedure to solve the problem optimally. Empirical testing based on the full theory is in progress.

In some of the database literature (e.g., Jin et al. 2003; Koudas et al. 2004), the assumption appears to be that the main goal of matching two files A and B is to obtain the set of pairs that are within epsilon using a nearest-neighbor metric or above a certain matching score cutoff. Although it is one of the fundamental issues in matching pairs of large files, it ignores how many true matches are within the set of pairs above a certain matching score and how many true matches are missed by the procedures for bringing together pairs. The discussion in this section deals with estimating false match rates within a set of pairs (or somewhat analogously *precision*) that is most easily accomplished with suitable training data in which the true matching status of a pair has been labeled. *Precision* is defined as $P(M \mid \hat{M})$ where $\hat{M}$ are the designated matches. The *false match rate* is $1 - P(M \mid \hat{M})$. Larsen (2005) and Winkler (2005) provide methods for parameter estimation are often intended for situations without training data (unsupervised learning). Methods for estimating the number of missed methods are covered in section 4.4. Figure 2 (section 3.5) illustrates the situation where the left 10% tail of the estimated curve of matches is compared with the truth (45 degree line) and the 10% right tail of the estimated curve of nonmatches is compared with the truth for three data sets. The close agreement is considerably better than Winkler (2002) that needs a small amount of training data and Belin and Rubin (1995) that holds in a narrower range of situations than Winkler (2002, 2005). The general research problem of estimating these error rates is still open because the current methods only apply in a narrow range of situations.

Estimating the proportion of false nonmatches (or somewhat analogously the complement of *recall*) is still an open problem. *Recall* is given by $P(\hat{M} \mid M)$. Even in the situations where there is representative training data, the problem may be difficult. The difficulty is due to the facts that businesses on two lists can have completely different names and addresses and that existing distributional models often assume that distributions of nonmatches are unimodal and have tails that drop off quickly. The proportion of false nonmatches due to pairs being missed by a set of blocking criteria is still an open problem (Winkler 2004).

4.2 **Auxiliary information for matching files A and B**

A bridging file is a file that can be used in improving the linkages between two other files. Typically, a bridging file might be an administrative file that is maintained by a governmental unit. We begin by describing two basic situations where individuals might wish to analyze data from two files. The following tables illustrate the situation. In the first case (Table 10), economists might wish to analyze the energy inputs and outputs of a set of companies by building an econometric model. Two different government agencies have the files. The first file has the energy inputs for companies that use the most fuels such as petroleum, natural gas, or coal as feed stocks. The second file has the goods that are produced by the companies. The records associated with the companies must be linked primarily using fields such as name, address, and telephone. In the second situation (Table 11), health professionals wish to create a model that connects the benefits, hospital costs, doctor costs, incomes, and other variables associated with individuals. A goal might be to determine whether certain government policies and support payments are helpful. If the policies are helpful, then the professionals wish to quantify how helpful the

policies are. We assume that the linkages are done in a secure location, that the identifying information is only used for the linkages, and that the linked files have the personal identifiers removed (if necessary) prior to use in the analyses.

Table 10. Linking Inputs and Outputs from Companies

Economics- Companies		
Agency A		Agency B
fuel	------>	outputs
feedstocks	------>	produced

Table 11. Linking Health-Related Entities

Health- Individuals		
Receiving Social Benefits		Agencies B1, B2, B3
Incomes		Agency I
Use of Health Services		Agencies H1, H2

A basic representation in the following Table 12 is where name, address, and other information are common across the files. The A-variables from the first A file and the B-variables from the second (B) file are what are primarily needed for the analyses. We assume that a record r_0 in the A might be linked to between 3 and 20 records in the B-file using the common identifying information. At this point, there is at most one correct linkage and between 2 and 19 false linkages. A bridging file C (Winkler 1999b) might be a large administrative file that is maintained by a government agency that has the resources and skills to assure that the file is reasonably free of duplicates and has current, accurate information in most fields. If the C file has one or two of the A-variables designated by A_1 and A_2, the record r_0 might only be linked to between 1 and 8 or the records in the C file. If the C file has one or two B-variables designated by B_1 and B_2, then we might further reduce the number of records in the B-file to which record r_0 can be linked. The reduction might be to one or zero records in the B file to which record r_0 can be linked.

Table 12. Basic Match Situation

File A	Common	File B
A_{11} , ... A_{1n}	Name1, Addr1	B_{11},...B_{1m}
A_{21} , ... A_{2n}	Name2, Addr2	B_{21},...B_{2m}
.		.
.		.
.		.
A_{N1} , ... A_{Nn}	NameN, AddrN	B_{N1},...B_{Nm}

Each of the linkages and reductions in the number of B-file records that r_0 can be linked with depends on both the extra A-variables and the extra B-variables that are in file C. If there are moderately high error rates in the A-variables or B-variables, then we may erroneously assume that record r_0 may not be linked from file A to file B. Having extra resources to assure that the A-variables and B-variables in the large administrative file C have very minimal error rates is crucial to successfully using the C file as a bridging file. Each error in an A- or B-variable in the large administrative may cause a match to be missed.

The research problems involve if it is possible to characterize the situations when a bridging file (or some type of auxiliary information) can be used for improving the basic linkage of two files. It is clear that a large population register C that covers the same populations as files A and B and that has many additional variables $C_1, \ldots, C_k$ might be of use. If a government agency has the resources to build certain administrative sources that represent the merger of many files, represents a superpopulation of the populations associated with many other files, has extra variables, and has been sufficiently cleaned, then the file from the administrative sources can be used for improving the matching of other sets of files. It can also be used in estimating the effect of matching error on analyses (considered in section 4.6 below).

4.3. **Approximate string comparator search mechanisms**

In many situations, we need to bring together information that does not agree exactly or character-by-character. In the database literature, Hjaltson and Samet (2003) have shown how records associated with general metrics for quantitative data such as string comparator distance can be imbedded in d-dimensional Euclidean space R^d . If the file size is N, then the intent of the embedding to reduce search times from $O(N^2)$ to O(N) or O(NlogN) in some situations such as those associated with photo archiving. Several fields of numeric data are used to describe characteristics of the photos. Jin et al. (2003) have shown how to take the basic string comparator metrics Φ associated with comparing strings having typographical error and use a corresponding embedding of each field into R^d where the dimensionality is chosen to be relatively small (often less than 20) and there is a corresponding Euclidean metric Φ'. Each field is associated with an R-tree in R^d and pairs of fields from two files are associated with pairs of R-trees that are searched. The searching is approximate because not all of the pairs of strings with distances $\Phi \leq c$ can be associated with pairs of strings $\Phi' \leq c'$ in R^d. Some approximately agreeing strings in the original space will not be found (Hjaltson and Samet 2003). The embedding from the original space to the new space can be computationally prohibitive in the dimensionality d is quite large (above 50). A research question is when can the combination of embedding and new decision rules associated with the R^d metrics Φ' improve matching and significantly reduce computation in comparison with naïve methods for bringing together all pairs.

Chaudhuri et al. (2003) have an alternative procedure in which they approximate edit distance with q-gram distances, create new indexes using an IDF (inverted document frequency) method with selected subsets of the q-grams to improve the efficiency of the search by three orders of magnitude in comparison with naïve search methods. Because Chaudhuri et al. (2003) made use of the frequency of occurrence of q-grams, their method improved over more naïve comparison methods such as edit distance. Baxter et al. (2003) have much cruder methods of creating new q-gram indexes that should be far easier to implement than the methods of Chaudhuri et al. (2003). The research questions are "When might it be suitable to use the methods of Chaudhuri et al. (2003) or Baxter et al. (2003)?"

BigMatch technology (Yancey and Winkler 2004) is designed to use very large amounts of memory that is available with many machines. With conventional matching software, each file is

sorted according to an individual set of blocking criteria prior to the files being passed through the matching software. If there are ten blocking passes, then each file is sorted ten times and passed through the matching software ten times. With Bigmatch software, file A is held in memory with associated indexes and file B never needs to be indexed or sorted. An initial research question is when are the methods of Yancey and Winkler (2004) more suitable than the methods of Jin et al. (2003) or Chaudhuri et al. (2003). The *q*-gram methods of Baxter et al. (2003) can be directly used to extend Bigmatch technology. BigMatch technology can be adapted to databases and is likely to be far more efficiently computationally than methods used by Hernandez and Stolfo (1995). Winkler and Yancey (2006) are investigating parallel versions of BigMatch technology that should yield significant speed increases in comparison with the basic uniprocessor BigMatch. The uniprocessor version of BigMatch processes approximately 130,000 pairs per second on faster machines.

One of the fundamental issues is whether one should consider methods that bring together all pairs from two files A and B or the subset of pairs in files A and B that sufficiently close according to a nearest-neighbor metric or above a matching score cutoff. Gravano et al. (2001) have methods that provably bring together all pairs that are close to each other. Koudas et al. (2004) and Guha et al. (2004) provide generalizations that the authors claim can efficiently be used in moderate size applications. If we use the example of Winkler (2004c) that considers the 10^{17} pairs from a 300 million record Census, ten blocking passes yield 10^{12} pairs that contain 99.5% of the true matches. The remaining matches in the residual set of pairs represent 1 match in 10^{11} pairs. Any strategy for getting the last 0.5% of matches (that have no 3-qrams in common – Table 9) would necessarily need to bring together hundreds or thousands of pairs for each true match and be subject to exceptionally high false match rates due to the poor quality of the weakly identifying information. In situations where it will be exceptionally difficult to obtain the last *x*% of matches, it might be best to have a method for estimating how many matches are missed as in Winkler (2004c). In situations, where the missing *x*% is quite small, it seems that the most suitable procedures for using the merged files would be to have a straightforward adjustment for the proportion of missed matches. Two research problems are how to accurately estimate the number of missed matches and to (somewhat) accurately estimate the effect of missed matches on analysis.

A closely related issue is whether the merged file A $\cap$ B is a representative subset of files A and B. If A $\cap$ B is representative subset of A or B, then an analysis on A $\cap$ B might be representative of an analysis on B in that it allows reproduction of an analysis in the sampling sense (Winkler 2006). Zadrozny (2004; also Fan et al. 2005) deals with the analogous issue of when a training sample is representative and does not induce bias. The research problem is whether a merged file is representative of a population in A $\cup$ B or subpopulation in A or B in the sense of producing an analysis without bias in key parameters. An obvious issue is that any matching error in merging files A and B can result in a 'dirty' sample set A $\cap$ B that might not be suitable for some (or most) analyses.

4.4 **Creating functions and metrics for improving matching**

To build intuition, we describe an elementary matching situation. During the re-identification of record r_{A1} from file A with fields (*geocode*, *Icode*, *Y*) with record r_{B2} with fields (*geocode*, *Icode*, *Y'*), we use a crude metric that states that the first two variables should agree exactly and the last variables *Y* and *Y'* should agree approximately. The field *geocode* may be a geographic identifier for a small region. The field *Icode* may be an identifier of the industry of the company. If *Y* and *Y*' are known to be in the tails of a distribution such as a high income or unusual situation, then we may be able to deduce that a range in which we can say *Y* and *Y'* are likely to be approximately the same. In some situations, we have a crude functional relationship *f(X) = Y*

that allows us to associate the *X*-variables with a predicted *Y*-variable that may be close to a corresponding *Y'*-variable. In these situations, we can think of the functional relationship $f(X) = Y$ and other knowledge as yielding a metric for the distance between *Y* and *Y'*. The variables *Y* and *Y'* can be thought of as *weak identifiers* that allow us to associate a record in the first file with one or more records in the second file.

The most interesting situation for improving matching and statistical analyses is when name and address information yield matching error rates in excess of 50%. Sometimes, economists and demographers will have a good idea of the relationship of the A-variables from the A-file and the B-variables from the B-file (Table 12). In these situations, we might use the A-variables to predict some of the B-variables. That is, $B_{ij} = Pred_j (A_{k1}, A_{k2}, ..., A_{km})$ where j is the j^{th} variable in the B-file and $Pred_j$ is a suitable predictor function. Alternatively, crude predictor functions $Pred_j$ might be determined during iterations of the linkage process. After an initial stage of linkage using only name and address information, a crude predictor function might be constructed using only those pairs having high matching weight. Scheuren and Winkler (1997) conjecture that at most two hundred pairs having high matching weight and false match rate at most 10% might be needed in simple situations with only one A-variable and one B-variable for a very poor matching scenario. The functions relating A- and B-variables can be quite crude. The intuitive idea is that each pair of A- and B-variables and their associations of particular ranges of each can drastically reduce the number of B-records that can be associated with each A record.

Michalowski et al. (2003) have done related work in which they use information from auxiliary files to improve matching. Koller and Pfeffer (1998), Getoor et al. (2003), Lu and Getoor (2003), Taskar et al. (2001, 2002, 2003) have provided extensions of Bayesian networks representing links between corresponding entities that also can be used for improving matching. In the context of microdata confidentiality, Winkler (2004a,b) has shown how to create additional metrics to improve matching for files that are masked to prevent re-identification or with files of synthetic data that are produced according to statistical models. The metrics depend on the distributions of the original and masked variables and known analytic uses of files. For microdata confidentiality, 0.5-2.0% re-identification rates are sufficient. For general administrative lists, we would likely need matching (analogously re-identification) rates in excess of 80% of the pairs that correspond to the same entities. The research question is "Under what circumstances can functional relationships and associated metrics be created between A-variables in file A and B-variables in file B to improve matching?"

To better illustrate concepts of functional relationships and associated metrics, we describe various methods of clustering and closeness in certain metrics such as nearest-neighbor metrics. Both Winkler (2002) and Scheuren and Winkler (1997) apply simple methods of one-variable types of clustering, knowledge about how far apart records will be according to the clustering, and how close a given cluster from one file will be to a given cluster in another file. The methods of clustering and analytic correspondences between a set of variables in cluster in one file to a set of variables in clusters in another file can often be obtained from understanding of analytic (or functional) relationships. If there are representative truth data, then functional relationships can be obtained via a nearest-neighbor relationship or via the functions of a hidden layer for neural nets. We do not need precise knowledge. With *k*-nearest-neighbor, a cluster from one file might be associated with a corresponding cluster in another file (according to the truth data). Any new records in files A and B can be associated via their corresponding clusters. If an A record is in a cluster that corresponds to a B record via the corresponding cluster, then the pair of records are weakly related according to the weak identifier corresponding to the corresponding cluster pairs. If we do multiple clustering in each pairs of files and create multiple pairwise clustering to achieve a number of pairwise functional relationships, then we better determine true matches based on the multiple clustering relationships. We note that the methods of Winkler (1989a, 1993; also 2002) allow us to account for the dependencies between multiple agreements. A somewhat analogous situation would be where we have two records that agree on the name 'John

Smith' and the date-of-birth '1965Mar21' for which we need additional information such as current address or income to make a better match determination. Torra also (2004) provides some methods that indicate how clustering can be used for matching.

4.5 **Link analysis**

Link analysis can refer to methods of connecting information within a file or across files. Any information that can be associated with an entity can allow the entity to be associated with other entities. We explain the elementary concepts in terms of person matching because the ideas can be considerably easier than the related concepts for business matching. In a single file, a *weak identifier* allows us to associate a record of an individual with other individuals. For instance, a household identifier allows us to associate the individual with other members of the household. A first name of John allows us to associate the record with other records having John in the first name field. If we have a date-of-birth 1955.03.21 of the form YYYY.MM.DD in the date-of-birth field, then we can associate the record with all records having year-of-birth 1955, month-of-birth 03, day-of-birth 21, or full date-of-birth 1955.03.21. A combination of weak identifiers such as name 'John Anthony Smith,' address '1623 Main Street, Springfield, Ohio,' and date-of-birth '1955.03.21' may allow us to uniquely identify the record associated with 'John Smith.' A subset of the weakly identifying information such as 'John Anthony Smith' and 'Springfield, Ohio' or 'John Anthony Smith' and '1955.03.21' may also allow us to uniquely identify the information associated with 'John Smith.'

In identifying the set of information with the entity 'John Smith' we make assumptions that there is sufficient redundant information to overcome typographical errors in some of the weak identifiers and the fact that some of the identifiers may be not current. For instance, the weak identifier 'John A Smith' and 'Sprinfeld, OH' may be sufficient for us to identify information with 'John Smith.' This is true even though the weak identifier 'John A Smith' does not have the full middle name 'Anthony,' 'Sprinfeld' has typographical error causing it to differ character-by-character from 'Springfield,' and 'OH' is the usual postal spelling abbreviation of 'Ohio.' The combination of 'John Anthony Smith' and '1955.03.21' may not be sufficient to identify the entity 'John Smith' because there are more than 30,000 individuals in the U.S. with the name 'John Smith.' This means that there is on average approximately 1.5 individuals with any given date-of-birth. On the other hand, the combination of a rarely occurring name such as 'Zbigniew Anthony Varadhan' and date-of-birth '1955.03.21' may be sufficient to uniquely identify the entity 'Zbigniew Varadhan.' If the name 'Zbigniew Anthony Varadhan' is sufficiently rare and does not contain typographical error, then it may be sufficient to uniquely identify 'Zbigniew Varadhan.' This indicates that, if we can make use of the relative rarity of weak identifiers in a file or in a population associated with a group of files, then we may be able to find better unique identifiers for a subset of the individuals.

We can make use of information that links individuals in households. Table 13 provides a listing of individuals that reside in two households at different addresses in different cities. Because the individuals are associated with surveys that represent two different time periods, the individuals may not be the same. The first individual in the second listed household is missing first name. We know that the surveys are two years apart. The middle initials differ on the second and fourth individuals. Although the identifying information differs slightly, we might still assume that the two households represent the same family. If we were creating a large population file by merging many lists than we might be able to conclude that John is the correct first name associated with the first individual. We would not be able to correct the middle initials with the second and fourth individuals.

Table 13. Individuals in Two Households in Two Cities at Different Time Periods

Household1		Household2	
John A Smith	45	blank A Smith	43
Mary H Smith	43	Mary A Smith	40
Robert A Smith	16	Robert A Smith	14
Susan M Smith	14	Susan N Smith	11

In this situation, an individual weak identifier is the information that the individuals reside in a household together allows us to link the individuals. In analogous situation, we might attempt to link individuals from two different households in different regions even in one file such as a population census. If we have an auxiliary population file representing all of the individual entities for a fixed time period, then we may be able to do the linkage even if more of the first names and ages are missing in Table 13. A research issue is whether it is possible to characterize when various types of very weakly identifying information such as membership in the same household or membership in the same subset of a file can be improved for improving matching.

Object identification can be considered a method of associating a large amount of information from a set of files with a given set of entities (or objects) that are considered to represent different individuals in a population (Russell 2001). For instance, Pasula et al. (1999, 2001, 2003) show how to apply the ideas to motor vehicle surveillance from viewing pictures along a freeway in California. Each camera will have several successive pictures of a set of cars. The time and dates of all the pictures are known. The object identification issue is taking the set of images from different cameras and associated them with individual vehicles. The information from each camera can be quite good because it includes several pictures of the same vehicle at a given point in time. They use the time and dates to reduce the number of images from a second camera. Characteristics such as color, shapes at various viewing angles, and other weak identifiers of a vehicle can be quickly determined from the video images. They perform object identification by creating a model that is computed off-line but used in real-time. Russell (2001) and Pasula et al. (1999, 2001, 2003) create a large graph in which that weakly connects enormous sets of images. To train the model, they need a moderately large amount of training data for which the truth is known. They use Markov Chain Monte Carlo (MCMC) to optimize the likelihood of the set of objects associated with the set of cars. Although MCMC is computationally prohibitive in these situations, more appropriate computational methods are being developed. An initial research issue is when these methods can be applied in a computationally tractable manner (during training or during classification). A second research issue is whether MCMC object identification methods can be effectively used to enhance other methods.

We have observed that we can use any of the weak identifiers to connect information within files or across files. The direct matching of lists of businesses is known to be difficult because of substantial name and address variations (Winkler 1995). Often the best information for improving the linkage of two files may be from a large population register. If the government has expended substantial resources in creating the register, then it can be used as a bridging file for linking the other two lists. If name and addresses variations are carried in the register, then that information can be used to improve the matching. As an extreme example, we might keep ten addresses associated with each entity in the register. Partial agreement on name and one of the addresses might be sufficient as a clustering technique (McCallum et al. 2000) that leads to more expensive computation on the other fields that might be used in determining the actual linkages of the entities.

Agreement information obtained from sources such as the Internet may give variants of names and addresses for businesses that allow improved linkages. The information would need to be placed in tables or files. This type of information is particularly helpful in matching if subsidiaries must be connected with parent corporations. If a unique customer identifier is available in a companies list associated with a company, then different name and address variations associated with the customer identifier may be available for improving matching in other files. Overlapping customer lists from two different files may help identify individual business entities.

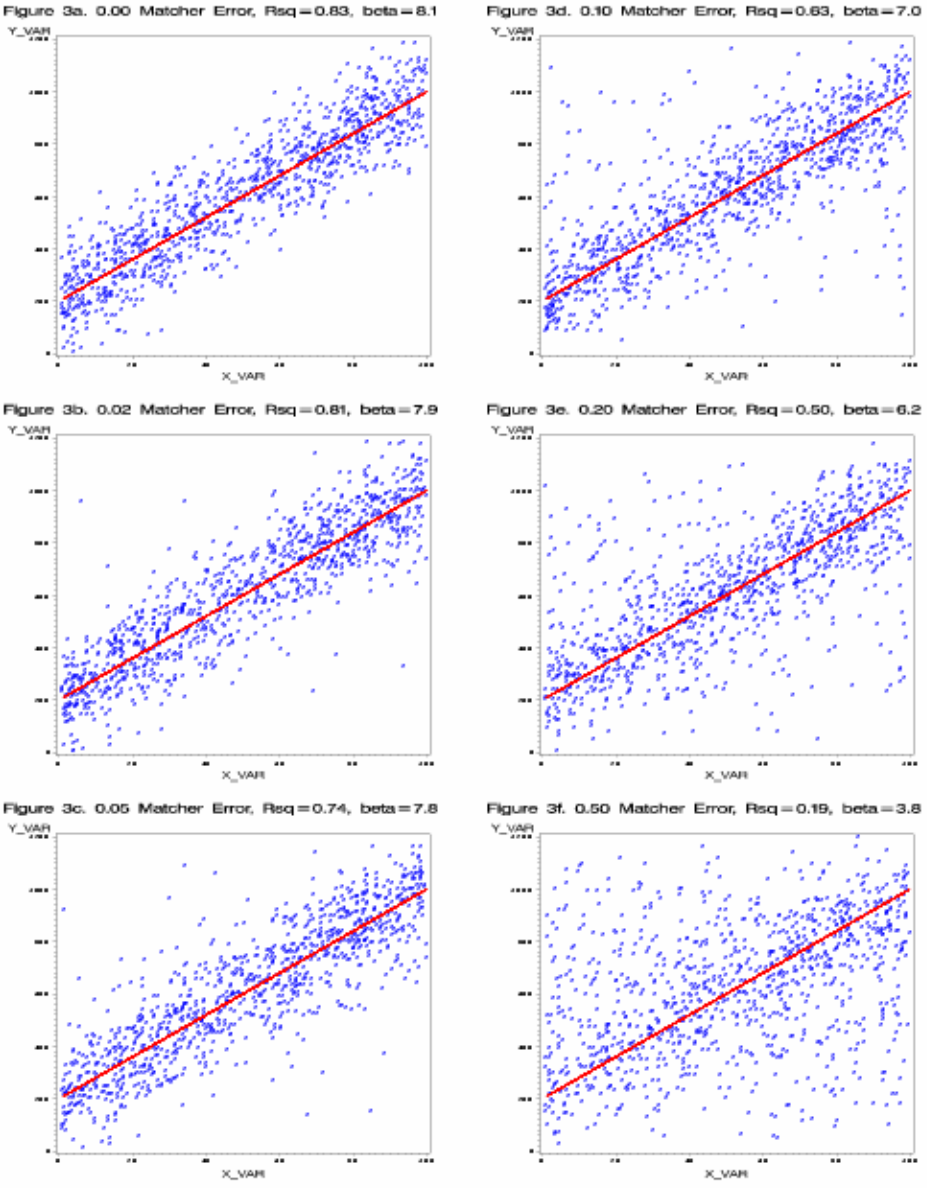
A key feature of the link analysis methods is that they provide a systematic way of bringing in information from other files. The methods are currently computationally prohibitive in most situations because they involve an enormous number of weak (or of lower quality) linkages via weak identifiers within and across files. Some of the weak identifiers such as functional relationships and associated linkage metrics can be constructed without truth decks. For many of the metrics associated with names and addresses, no truth decks are often needed. Combinations of weak identifiers or weak identifiers combined with other information can yield substantial improvements in matching accuracy.

At the most sophisticated level of modeling, a link analysis model can represent an enormous number of linkages that must be considered. Via an optimization of likelihood, the set of distinct objects and the linkage probabilities associated with their weakly identifying information can be determined (Lafferty et al. 2001). Ishikawa (2003) has provided a more general characterization of the set of linkages associated with link analysis. The set of linkages are a large graph representing a Markov Random Field. Ishikawa's methods may yield more computationally tractable methods of optimizing the likelihood. McCallum and Wellner (2003) provide a number of graph partitioning algorithms that can be computationally tractable in some situations. Ravikumar and Lafferty (2004) provide methods that show promising improvement for the theory and computation associated graph partitioning and belief propagation.

4.6 **Adjusting Analyses for Linkage Error**

The purpose of merging pairs of files A and B is often the situation of Table 12 in which we are primarily interesting in analyzing the joint relationships between the A-variables and the B-variables. Scheuren and Winkler (1993) considered the simple situation where a continuous x-variable variable was taken from the A file and compared with the continuous y-variable from the B-file. Among true matches, the pair of variables satisfied the regression relationship $y = \beta x$ where the R^2 value was above 0.4. Among false matches, an x-value would typically be associated with a y-value that had been randomly drawn from the entire range of y values. Scheuren and Winkler (1993) provided a bias-adjustment procedure for the estimate of the beta coefficient that was generalized by Lahiri and Larsen (2005). Figure 3 illustrates the effect of increasing matching error rate on the point cloud associated with a simple regression of the form $y = \beta x$. The true regression line is shown in each component of the graph. As matching error increases from Figure 3b until Figure 3f, the regression relationship almost disappears.

Further, Scheuren and Winkler (1997) showed how to provide a predicted value pred(y) = βx became an additional matching variable (beyond name and address) that could be used in matching. The predicted value was equal to observed y-value when the observed y-value was within two standard deviations of the regression error; else it was equal the predicted value βx. This simple addition of one additional matching variable reduced the matching error rate from well above 50% to a matching error rate of 10%. With a larger number of pairwise relationships between additional correlated variables from the files A and B, Scheuren and Winkler conjectured that the matching accuracy could further be improved. The improvement in matching efficacy is



analogous to the improvement in having a third matching variable of current address in addition to two matching variables consisting of name and date-of-birth. Another analogous situation is where a given company can be associated with another company via postal ZIP code and industrial NAIC code. If one record in one file has an income variable and another record in

another file has a receipts variable, then a simple set of piecewise regression relationships can yield improved matching (Steel and Konshnik 1999).

If we are interested in an (X, Y) relationship where X taken from file A is multivariate and Y taken from file B is multivariate, then we might be able to use a mixture model approach for partially improving the matching. As with the Scheuren and Winkler (1997), we separate (X_T, Y_T) pairs associated with true matches from (X_F, Y_F) pairs associated with false matches using some of the known analytic relationships known to hold among (X_T, Y_T) pairs and the fact in (X_F, Y_F) pairs the Y_F value in drawn randomly (in many or most situations) from the entire set of Y values. If the rate of false matches is high, then a mixture model approach may help to separate (X_T, Y_T) pairs from (X_F, Y_F) pairs. The modeling of relationships can be refined in the set of (X_T, Y_T) pairs that have the highest probability of being correct, new relationships and associated predicted values added to one file to be used as additional matching variables, and matching improved. One research problem is determining a number of situations when we can use these types of procedures to improve matching. A closely related problem is when is it possible to use the procedures to improve a specific analysis?

4.7 **Relative Frequency, Typographical Error Rates, and Overlap of Lists**

Chaudhuri et al. (2003) introduced methods for probabilistically approximating (using Chernoff bounds) edit distance with distances based on *q*-grams, methods that used the relative frequency of rarer (less frequently occurring) *q*-grams, and some clever heuristics in creating a matching strategy for locating probable duplicates within a database. The *q*-grams are used in a type of inexpensive clustering procedure for bringing together pairs (McCallum et al. 2000, also Ravikumar and Cohen 2004) and for computing matching scores that are based on the relative frequency of strings. The only facet of their work that we address is the relative frequency of strings used in matching, a notion that was introduced by Newcombe et al. (1959, 1962) and given a somewhat formal model by Fellegi and Sunter (1969) and Winkler (1989c). In particular, we will consider a number of situations where use of relative frequency of strings is will not improve matching (Winkler 1989c, Yancey 2000).

The work of Newcombe et al (1959, 1962) and later closely related work by Gill (1999) primarily depended on a large, complete population file that could be compared against itself in developing appropriate relative frequency tables. The intuition is that a relatively rarer last-name string such as 'Zabrinsky' has more distinguishing power than a string such as 'Smith.' In using the frequency tables from the large population file, a smaller file can be matched against the large file under the plausible assumption that the smaller file is an approximate subset of the larger file. Fellegi and Sunter (1969) dealt with the more general problem where two files A and B were not assumed to be approximate subsets of a large population file for which relative frequencies needed to be computed under some strong assumptions on the conditional independence of different fields. Winkler (1989c) significantly weakened the strong assumptions by scaling the frequency weights to certain conditional probabilities that had been computed under (unsupervised) EM procedures. Chaudhuri et al. 2003) also based their matching scores on frequencies of strings that had been computed by matching a file against itself.

Accounting for relative frequency in a rigorous manner is difficult for several reasons. The first is that two files A and B may not overlap much. For instance, if file A is a file of physicians and file B is an incomplete listing of the general population, then agreement on a rarer name such as 'Zabrinsky' may help significantly more than agreement on a frequent name such as 'Smith.' Deciding on a match will depend more on the additional fields beyond last name that are used in the matching. The second reason is that if there are much higher typographical error rates in a last name such as 'Zabrinsky.' then agreement on 'Zabrinsky' may depend more on whether typographical errors have occurred and relative frequency tables will be inaccurate. The third reason is that, if a smaller list A is compared to a larger list B, then the relative frequencies can

again be inaccurate. For instance, if A is a 0.1% sample of a larger population file and B is a 0.1% sample of the same larger population, then the relative frequency of a rare name such as 'Zabrinsky" that happens to be sampled into files A and B may be approximately as high as a common name such as 'Smith.' The research problem is whether relative frequency can be effectively accounted for in a model that improves matching efficacy. Yancey (2000) has demonstrated that relative frequency matching does not improve on simple yes/no matching in small geographic regions when there are a number of other high quality fields for matching.

Cohen et al. (2003a,b) introduced a soft TFIDF weight as an enhancement to the TFIDF weighting of information retrieval that is often used in comparing documents and in search engines. Part of the TFIDF weighting has strong similarity to the value-specific (frequency-based) weighting of Newcombe (1959, 1962). The TFIDF weight, or *cosine similarity*, is defined as

$$TFIDF(A,B) = \sum_{w \in A \cap B} V(w,A) \cdot V(w,B)$$

where $TF_{w,A}$ is the frequency of word *w* in A, N is the size of the file A $\cup$ B, IDF_w is the inverse of the fraction of names that contain *w*,

$$V'(w,A) = \log(TF_{w,A} + 1) \cdot \log(IDF_w) \text{ and}$$

$$V(w,A) = V'(w,A) / \sqrt{\sum_{w'} V'(w,A)}$$

where the sum is over the entire set of words. In this situation, the words *w* can be any strings such as first names, last names, house numbers, street names, etc. that are used in the matching. In the situations of classical record linkage (Winkler 1989c), individual fields (words) such first names, last names, etc. are scaled very differently. To deal with typographical error, Cohen et al. (2003a,b) introduced

$$SoftTFIDF(A,B) = \sum_{w \in CLOSE(\theta,A,B)} V(w,A) \cdot V(w,B) \cdot D(w,B).$$

Here the set $Close(\theta, A, B)$ represents the set of words (strings) in A that are close to words (strings) in B with a suitable string comparator. Cohen et al. used the Jaro-Winkler string comparator with θ=0.9. *D(w, B)* is the closest (i.e., highest string comparator weight) word in B to word *w*. There are issues with the methods of Cohen et al. (2003a,b). The first is whether their thirteen small test data sets are sufficiently representative of general data in terms of typographical error and relative frequencies. The typographical error affects the computation of TFIDF weights and analogous frequency-based weights in Winkler (1989c). If there is one last name 'Smoth' that represents a typographical error in last name 'Smith,' then the weight associated with 'Smoth' will be too high. The adjustment using *D(w, B)* is likely to be difficult to determine with larger real world data sets because of the associated amount of computation. Winkler (1989c) had a general downweighting based on the matching weights $P(f_i^a \mid M)$ that were computed via the EM algorithm. The advantage of the Winkler (1989c) downweighting adjustment for typographical error is that it is far faster (but cruder) than the method of Cohen et al. (2003a,b). Based on early empirical evidence, it seems likely that the method of Cohen et al. will work well in practice. A research problem is determining the situations when the Cohen et al. method will be superior to other simpler and less compute-intense methods.

Cohen et al. (2003b) introduced the idea of a modified Levenstein metric that worked well in a number of situations. The basic Levenstein metric measures the number of insertions, deletions, and substitutions to get from one string to another. A standard rescaling for consistency is to divide the Levenstein metric by the total number of unique characters in two strings and subtract the ratio from 1. Cohen et al. (2003b) modified the Levenstein metric by adjusting the 0-1 scaled metric upward for exact agreement on the first few characters of the strings. They found that the rescaled Levenstein metric worked best in some matching situations. A research issue is on what types of files the rescaled Levenstein metric can be expected to yield improvements over other metrics. For improving record linkage in general, an important research issue is having a number of large, representative test decks for which true matching status is known. Although some researchers (Hernandez and Stolfo 1995, Chaudhuri et al. 2003, Scannipieco 2003) have used artificially generated files in which 'typographical' errors are induced in true values of certain strings, the generated data does not approximate the types of errors that occur in real data. Extreme examples occur in Table 9 and in most business lists.

A general issue is whether it is possible to create a model that characterizes typographical error in a manner that it allows estimation of the number of matches or duplicates that are obtained with a given set of blocking strategies. Historically, blocking strategies have been determined using trial and error (Newcombe 1988, Gill 1999, Winkler 2004c). Chaudhuri et al. (2003) and Ravikumar and Cohen (2004) used inverted indexes based on the q-grams for bringing together pairs for which a much more compute-intensive procedure was used for computing matching scores. A real issue is that the q-grams used in the inverted indexes may bring together far too many pairs in the sense that the set of pairs contains an exceptionally small proportion of matches. A similar issue can occur with blocking criteria. For instance, if we are matching a ZIP code of 50,000 individuals against itself, at most one in 50,000 pairs will be a true match. Different blocking passes are intended to keep the number of pairs brought together relatively smaller. For instance, three blocking criteria might be (1) ZIP plus the first few characters of last name, (2) ZIP plus the first few characters of street name, and (3) part of last name, first name and the street address. The strategy would partially account for some of the typographical error in last name, street name, and ZIP code. Generally, it will not bring together all of the pairs that might have a matching score (based on all fields in the pair of records) above a certain cutoff score.

Winkler (1989b, 1995, 2004c) gave a capture-recapture method for estimating the number of matching pairs that are missed by a set of blocking criteria. The estimation method can be inaccurate due to the correlation of captures of pairs across blocking criteria and due to the inaccuracy in estimating matching error rates. Fellegi and Sunter (1969) and Winkler (1989c) gave methods for estimating error rates in individual fields that might be adapted to groups of fields. Winkler (2004c) provides methods for estimating typographical error rates and the approximate distinguishing power fields used in searches. Although the estimates are quite inaccurate, they do not need training data and can be used to provide crude upper bounds. Winkler (2004c) also provides a very large application of sets of blocking criteria to the estimation of the number of matches missed by a set of blocking criteria. The reason that the problems are particularly difficult is that the typographical errors typically occur multiple times within a record or within the records associated with a household. The errors do not apply in isolation and are not independent. Nevertheless, it is possible to get crude bounds on the number of missed matches. It is noteworthy that the most-difficult-to-find 0.2-0.5% of matches have no 3-grams in common. These matches are often identified because some other individual in a household is identified more easily (Table 9). The research problem is when these capture-recapture methods can be used to get accurate estimates of the proportion of missed matches or at least reasonable upper bounds of the proportions. A related problem is if it is possible to adjust an analysis in the linked file for the proportions of missed matches.

4.8 **More Fields, Lack of Independence, and Typographical Error Rates**

Most individuals assume that having more information (fields) for matching would improve the matching. Additional fields only help in certain situations. Generally, having more fields than 6-10 fields for matching is not needed. In this section, we illustrate the situation with a number of examples. To begin we need to describe a particular notion of a typographical error rate and how matching weights are computed under conditional independence (naïve Bayes) and in general. Let an agreement pattern have the simple yes/no form {0/1, 0,1, …., 0/1} where the first entry is agree/disagree on the first field f_1, the second entry is agree/disagree on the second field f_n, …, and the nth entry is agree/disagree on the nth field f_n. In situations where there is very little transcription and keypunch error, we expect matches in M to agree on most fields and nonmatches in U to only randomly and occasionally agree on fields. There are obvious exceptions where a husband/wife nonmatch agrees on many characteristics such as last name, age, house number, street name, telephone, etc. Most nonmatches will be almost completely unrelated in the sense that few fields agree. There may be occasional random agreements on first names, last names, ages, or street name.

In computing the matching score (weight) in equation (1) under the conditional independence assumption, we obtain

$$\log[\frac{P(f_1^x, f_f^s, ..., f_n^x \mid M)}{P(f_1^x, f_2^x, ..., f_n^x \mid U)}] = \sum_{i=1}^{n} \log[\frac{P(f_i^x \mid M)}{P(f_i^x \mid U)}] \qquad (4)$$

where f_i^x represents either agree a or disagree d on the ith field. Typically, $P(f_i^a \mid M) > P(f_i^a \mid U)$ and $P(f_i^d \mid M) < P(f_i^e \mid U)$. This causes the agreement weights (log ratios) associated with individual fields to be positive and the disagreement weights to be negative. If one field has very similar characteristics to another, then agreement on the additional field may not be helpful. For instance, if f_1^a is agreement on last name (surname) and f_2^a is agreement on Soundex code of surname, then $P(f_1^a \mid C) = P(f_1^a, f_2^a \mid C)$ where C is either M or U. If we compute the total agreement weight on the left hand side of (4) using the individual agreement weights on the right hand side, the total agreement weight will include $\log(P(f_2^a \mid M) / P(f_2^a \mid U)$ that erroneously increases the total agreement weight. The erroneous increase is primarily due to the fact that the conditional independence assumption does not hold.

In actuality, $\log(P(f_2^a \mid M) / P(f_2^a \mid U)$ should be set to 0. If we compute the left hand side of (4) using the right hand side, then any field f_j that is completely dependent on the several other fields should not be computed according to the conditional independence assumption in (4). The erroneously computed weights can affect the classification rule (1). As an additional example, if f_1 is last name, f_2 is date-of-birth, f_3 is US Postal ZIP code+4 (geographic region of approximately 50 households), and f_4 is first name, then agreement on f_1, f_2, f_3 in M assures agreement with probability one on f_4. In this situation, the numerator on the left hand size of equation (4) would be too low if it were computed under conditional independence, the denominator might stay the same, and the overall weight given by the left hand size of (4) would be too low. Winkler (1989a, 1993) and Larsen and Rubin (2001) have provided ways of dealing with lack of independence.

Fellegi and Sunter (1969) and Winkler (1988, 1989c) considered $P(f_i^a \mid M)$ as an approximate measure of typographical error. Winkler showed that the EM-based parameter estimation

procedures could yield reasonable estimates of $P(f_i^a \mid M)$ for use in the matching classification rule (2). Here a *typographical error* might represent any difference in the representation of corresponding fields among matches within a pair of records. For instance, first name pairs such as (Bill, William), (Mr, William), (William, James), or (William, Willam) might all represent a typographical error. In the first example, the nickname Bill corresponds to William, in the second example, Mr may represent a type of processing error, in the third example, and James may be a middle name that is in the first name position. In the fourth example, the misspelling Willam could be dealt via Jaro-Winkler string comparator ρ by having strings with ρ > 0.89 as agreement and disagreement, otherwise.

There are research issues associated with computing agreement weights where there are possible dependencies and varying amounts of typographical error. In the first situation, the EM algorithm can provide reasonable estimates of the marginal probabilities $P(f_i^a \mid M)$ and $P(f_i^a \mid U)$ where the probabilities yield good classification rules using (2). If the probabilities yield good classification rules, then $P(f_i^a \mid M)$ may represent reasonable surrogate measure for typographical error. In many situations, if a typographical error occurs in one field in a record, then it is more likely to occur in another field of a record. This means that while dependencies between agreements on fields may increase or decrease the matching weights in (4), the dependencies between typographical errors have a tendency to decrease the matching weighs in (4). The research question is "How does one estimate probabilities in equation (4) that yield good classification rules (or estimates of error rates) when there are dependencies between fields and dependencies between typographical errors in fields?" If there are many fields, then how does one select a subset of the fields that provide a suitable amount of distinguishing power in terms of the classification rule (2). Certain feature selection procedures (e.g., Sebastiani 2002) may yield suitable subsets of fields.

4.9 **Algorithms Used in Record Linkage Classification Rules**

The standard model of record linkage is conditional independence (naïve Bayes) that has been extended to general interaction models by Winkler (1989a, 1993) and Larsen and Rubin (2001). In machine learning, support vector machines (SVMs, Vapnik 2000) and boosting (Freund and Schapire 1996, Friedman et al. 2000) typically outperform naïve Bayes classifiers and other well understood methods such as logistic regression. In this section, we describe the four classifiers in terms of theoretical properties and observed empirical performance. We begin by translating the notation of the four classifiers into a vector space format. If $r = (f_1, \ldots, f_n)$ is a record pair of n fields used for comparison and $(a_1, \ldots, a_n)$ are agreement vectors associated with the n fields, then, in the simplest situations, we are interested in a set of weights $\boldsymbol{w} = (w_1, \ldots, w_n)$ such that

$S(r) = \sum_{i=1}^{n} w_i a_i > C_H$ means the pair is a match (in one class),

$S(r) = \sum_{i=1}^{n} w_i a_i > C_L$ means the pair is a match (in other class), and

otherwise, the pair is held for clerical review. In most situations, $C_H = C_L$ and we designate the common value by C. In this section, we assume that representative training data is always available. In logistic regression, we learn the weights $\boldsymbol{w}$ according to the logistic regression paradigm. In SVM, we learn an optimal separating linear hyperplane having weights $\boldsymbol{w}$ that best separate M and U (Vapnik 2000). With N steps of boosting, we select a set of initial weights $\boldsymbol{w}^o$ and successively train new weights $\boldsymbol{w}^i$ where the record pairs r that are misclassified on the

previous step are given a different weighting. The starting weight $\boldsymbol{w}^0$ is usually set to *1/n* for each record where *n* is the number of record pairs in the training data. As usual with training data, the number in one class (matches) needs to be approximately equal the number of pairs in the other class (nonmatches). Ng and Jordan (2002) and Zhang and Oles (2001) have demonstrated that logistic regression can be considered an approximation of SVMs and that SVMs should, in theory, perform better than logistic regression.

Ng and Jordan (2002) have also demonstrated empirically and theoretically that SVM-like procedures will often outperform naïve Bayes. Various authors have demonstrated that boosting is competitive with SVM. In record linkage under conditional independence, each weight $w_i = P(f_i^a \mid M) / P(f_i^a \mid M)$ for individual field agreements are summed to obtain the total agreement weight associated the agreement pattern for a record pair. The weighting of this type of record linkage is a straightforward linear weighting. In theory, SVM and boosting should outperform basic record linkage (possibly not by much) because the weights $\boldsymbol{w}$ are optimal for the type of linear weighting used in the decision rule. One reason that SVM or boosting may not improve much is that record linkage weights that are computed via an EM algorithm also tend to provide better separation than weights computed under a pure conditional independence assumption (Winkler 1990a). Additionally, the conditional independence assumption may not be valid. If conditional independence does not hold, then the linear weighting of the scores *S(r)* is not optimal. Alternate, nonlinear methods, such as given by Winkler (1989a, 1993b) or Larsen and Rubin (2001) may be needed.

There are several research issues. The first issue is to determine the situations where SVM or boosting substantially outperforms the Fellegi-Sunter classification rule (2). Naïve Bayes is known to be computationally much faster and more straightforward that SVM, boosting, or logistic regression. Belin and Rubin (1995) observed that logistic regression classification is not competitive with Fellegi-Sunter classification. The second issue is whether it is possible to develop SVM or boosting methods that work with only unlabelled data (Winkler 1988, 1993) or work in a semi-supervised manner as is done in record linkage (Winkler 2002). Brefeld and Scheffer (2004) provide a semi-supervised version of SVM. Extensions of the standard Fellegi-Sunter methods can inherently deal with interactions between fields (1993) even in unsupervised learning situations. Determining the interactions can be somewhat straightforward for experienced individuals (Winkler 1993b, Larsen and Rubin 2001). SVM can only be extended to interactions via kernels that are often exceeding difficult to determine effectively even with training data. Fellegi-Sunter methods can deal with nearly automatic methods for error rate (false match) estimation in semi-supervised situations (Winkler 2002) or in a narrow range of unsupervised situations (Winkler and Yancey 2006). Accurately estimating error rates is known as the regression problem that is very difficult with SVM or boosting even when training data are available (Vapnik 2000, Hastie et al. 2001).

4.10 String Comparators and Extensions

The basic idea of string comparison is to be able to compare pairs of strings such as ‘Smith, Snith’ that contain minor typographical error. Winkler (1990a) showed that even high quality files might contain 20+% error in first name pairs and 10+% error in last name pairs among pairs that are true matches. He demonstrated that being able to account for the minor typographical error could significantly improve matching efficacy because a (sometimes substantially) higher proportion of true matches could be found automatically. With a priori truth decks he was able to model the effect of minor typographical error on the likelihoods (equation (1)) used in the main classification rule given by equation (2). He observed that the two initial variants of the string comparator yielded significant improvements in matching. Further variants of the two string comparators and the likelihood-ratio-downweighting function yielded little or no improvement in

the types of Census files that he used for the empirical applications. The minor typographical error rate can significantly increase with files that have been scanned from handwritten forms (Winkler 2004c). In those situations, having effective string comparator functions is crucial.

The Jaro, Jaro-Winkler, and edit-distance string comparators provide fast, effective methods of comparing strings having typographical. At present edit distance appears to be one of the most effective off-the-shelf string comparators (Cohen et al. 2003a,b). Edit distance measures the minimum number of insertions, deletions, and substitutions to get from one string to another. Each insertion (ε, *b*), deletion (*b*, ε), and substitution (*a*, b) is given cost one in the dynamic programming algorithm used to obtain edit distance. Here ε is the null character. Characters *a* and *b* are arbitrary characters. With ordinary lower-case alphabetic characters plus the blank (space) character and null character ε, there are 27 possible values for each character. For speech recognition, Ristad and Yanilios (1998) introduced hidden Markov models in a likelihood-based computation of a generalization of edit distance. They assumed that sufficient training data would be available and that costs of specific substitutions *(a*, *b*), insertions (ε, *b*), and deletions (*b*, ε) over different characters *a* and *b* could be modeled and estimated. Ristad and Yanilios demonstrated how to use the Viterbi algorithm (fast version of EM algorithm) for computing the generalized edit distance.

Yancey (2004) examined Markov edit-distance generalization of Wei (2004) and concluded that the new metric was slightly outperformed by the Jaro-Winkler string comparator with certain types of test decks. In a more ambitious study, Yancey (2006) compared a number of variants of several types of string comparators. His best general variant of the Jaro-Winkler comparator, called JW_{110}, performed a prefix enhancement and applied an enhancement for string similarity. Yancey (2006) felt that the third potential enhancement of a suffix comparison performed yielded improvements in some situations and performed worse in others. His best alternative string comparator *LCSLEV* was from a simple averaging of the longest common subsequence string comparator with the basic edit (Levenshtein) distance. Each of these string comparators was converted to the real-number range between 0 and 1 range prior to averaging. His final hybrid comparator was a nontrivial function of the scores of the JW_{110} and *LCSLEV* string comparators. The hybrid comparator very slightly and consistently outperformed both JW_{110} and *LCSLEV* in matching tests with several test decks. The metrics in the tests were the number of matches obtained at given false match rates.

Bilenko and Mooney (2003a) argued that Hidden Markov models could be used for training the comparison metrics for individual fields and that other algorithms such as SVMs could be used in finding the optimal set of SVM weights for separating matches from nonmatches. The advantages of SVMs are that they are known empirically to perform very well and several SVM implementations are freely available. Similar Hidden Markov algorithms have been used for address standardization. In speech recognition and in simple edit-distance tasks, we know the alphabet for the characters. In more general situations where words are compared, we need smoothing methods (Bilenko ad Mooney 2003a) to be able to weight (i.e, compare) two strings in which one or both have not been previously encountered.

The research issues are somewhat straightforward when there are training data. Bilenko and Mooney (2003a) claim that, if suitable training data, then their method will optimize the comparison of individual fields and the overall set of fields between two records. An initial research question is when their methods will yield improved matching performance in comparison with much simpler methods such as those based on a Fellegi-Sunter model of record linkage. A second issue is when Bilenko-Mooney methods will be suitable if there are very small or no amounts of training data. Winkler (1988, 1989a, 1993) and Ravikumar and Cohen (2004) provided methods for matching that did not need training data. The Ravikumar-Cohen methods, while less general than Winkler (1993), were computationally much faster and possibly suitable for a large number of matching situations. Yancey (2004) made use of large training decks for

learning the generalized edit distances. The idea was that the generalized edit-distance would be used with the likelihood ratios of equation (2) in a manner similar to that used by Winkler (1990a). The research issue is when can the generalized edit-distance produce clearly superior results to those produced with methods such as Jaro-Winkler string comparators? A third research issue is subtle and very difficult. Winkler (1989a) demonstrated that optimal matching parameters vary significantly across adjacent geographic regions. In particular, because of higher typographical error rates in urban regions, urban regions might not be matched as accurately as adjacent suburban regions. If there is significant variation of typographical error across geographic regions, do different generalized edit-distances need to be learned for each region? If that is the situation, how much representative training data is needed for each geographic region?

4.11 **Standardization and Data Extraction**

In record linkage, *standardization* refers to methods for breaking free-form fields such as names or addresses into components that can be more easily compared. It also refers to methods for putting dates such as 12 December 2005 or Dec. 12, 2005 into a standardized MMDDYYYY format of '12122005' or sex codes such as '0', '1,' and '2' into more easily maintained codes of blank (' '), male ('M'), and female ('F''). Standardization is not easy even in seemingly straightforward situations. If we know that free form names in a particular file are supposed to have first name first, then we may be able to parse the name into components such as first name 'William' and last name 'Smith' as illustrated in lines 1-3 of Table 14. Name standardization software (e.g. Winkler 1993a) looks for certain prefix words such as Rev, Dr, Mr, etc, certain postfix words such as MD, PhD, etc and then breaks out remaining words into first name, middle initial, and last name. Last names having certain typical prefixes such as 'Van,' 'De,' are also easily handled. After putting in standard spelling for words such as Reverend, Doctor, etc, the software generally breaks the words into components that are assigned a pattern that is looked up in a table that determines the final parsing of the free-form name. The tables and overall logic are determined using large test decks with typically encountered free-form names. No current software can deal with the situation where first names such as 'Stanley' and switched with last names such as 'Paul' as shown in lines 4-5 of Table 14.

Table 14 Examples of Free-form Names

1. Dr. William E. Smith, M.D.
2. William Edward Smith, MD
3. Dr. Willam F. Smith
4. Paul Stanley
5. Stanley Paul

The methods for name standardization described above are referred to as 'rule-based' because the rules for converting a free-form name into components are fixed. No probability models are involved. Generally, the fixed rules are developed with a very large, representative test deck in which the forms of the name components are known because they have been manually processed by experts. At present, I am unaware of any readily available commercial software for name standardization. Because of the large market for address software, there is readily available, excellent commercial software. Typically, commercial software must meet minimal standards based on testing against large U.S. Postal Service test decks to be certified. The logic of the commercial address standardization software (like comparable software from the Geography Division of the U.S. Census Bureau) uses rule-based methods that are often much faster than probability-model-based methods.

The probability methods based on Hidden Markov models introduced by Borkar et al. (2001) and also applied by Churches et al. (2002) and Christen et al. (2002) show considerable promise for certain types of address standardization. Borkar et al. (2001) applied a 2nd order HMM model that required sophisticated modifications in the Viterbi algorithm (very fast type of EM algorithm for specific types of applications) that did not require lexicons at initial levels. Churches et al. (2002) applied a simpler 1st order HMM that used lexicons to initially tag words such as possible female first name, possible female last name, possible surname, type of postal address, locality (town, city) names, etc. A key feature of the Hidden Markov models is the ability to create additional training examples (with assistance from the software) as new types of data files are encountered (Churches et al. 2002). It is relatively straightforward for a human being to parse new types of addresses into components that can be used as new training examples. The 2nd order Hidden Markov models work well with the types of unusual address patterns encountered with Southeast Asian and Indian addresses. Both 1st and 2nd order HMMs work well with Western style addresses. As Churches et al. (2002) and Christen et al. (2002 note, rule-based methods work well with Western style (house-number / street-name) that are often used in North America, Western Europe, and Australia. The 1st order Hidden Markov methods, however, do not presently perform as well as the rule-based methods for name standardization.

There are a large number of situations where probability-types of models can be used for efficient preprocessing of data. The methods often apply Hidden Markov models and other probabilistic methods such as Conditional Random Fields. The methods arose in extremely difficult data-extraction situations such as Internet web page comparison or general comparison of sentences in natural language processing. Cohen and Sarawagi (2004) apply dictionaries (lexicons) in named entity extraction that yield improvements in comparison with more basic methods of name or address standardization. Their methods might yield extensions or alternatives to the 1st and 2nd order HMM models of Borkar et al. (2001) and Churches et al. (2002), respectively.

Agichstein and Ganti (2004) provide alternate methods of segmenting text that depend on having a large reference file in which each of the respective fields have been cleaned. As an instance, we might have a person record with a free-form name, free-form address, and other information. In the person record, we wish to identify the location of the name information and the location of the address information and divide each of the respective fields into components that can be more easily compared. As another instance, we may have a reference to a journal article that is in the form (author, date, title, journal, volume number, pages) where we want to identify each of the components and put them in an appropriate. No specific training data is needed. For their CRAM (Combination of Robust Attribute Models) system, Agichstein and Ganti (2004) create the needed information and dictionaries (lexicons) from the reference file.

CRAM is based on an *Attribute Recognition Model* that uses information from individual fields in the reference file to locate different fields in the file being processed. The idea is that a name field, address field, or date-of-birth field in the input file being processed will look like corresponding fields in the reference file. CRAM creates a specific attribute recognition model (ARM_i) for each field F_i or attribute A_i in the reference file. Given an input string s, CRAM partitions s into s_1, …, s_n, maps the segments s_i to distinct attributes A_{si}, i = 1, …, n, such that $\prod_{i=1}^{n}\{ARM_{si}(s_i)\}$ is maximized over all valid segmentations of s into n substrings.

CRAM is trained using a Hidden Markov Model (described in section 4.10), has a feature hierarchy to help recognize tokens that have not been encountered previously, contains a generalized dictionary to all elements in the feature hierarchy and set of base tokens. The string s = [Amazon Company, 1243 Main Street, Seattle, WA, 92111] has initial token 'Amazon Company' and token set {amazon, company}. Each ARM_i is divided into *Beginning, Middle*, and *Trailing* attribute models. The dictionary D_i corresponding the i^{th} attribute is divided in beginning,

middle, and trailing dictionaries D_i^B , D_i^M, and D_i^T . Agichstein and Ganti (2004) introduce substantial enhancements, needed computational relaxations, and robustness.

4.12 **Using Multiple Addresses Effectively**

Population registers or large files (denoted by file C) that contain extra information may be very useful for improving the matching of two files A and B. A government agency or other group may have the resources to maintain a file C. In addition to names that have been cleaned up and put in a consistent format, file C may contain five of the most recent addresses associated with an individual. Each address has a date (vintage) associated with it. File C is updated with the most recent addresses and the oldest previous address is dropped. Assume that File A has an entry 'John Smith 123 Main St' and File B has an entry 'John Smith 456 Oak Ave.' It may be possible to determine accurate match status if file C contains both addresses '123 Main St' and '456 Oak Ave' associated with some individual John Smith. Other weak identifiers might be previous phone numbers, places where someone has purchased products, or places where someone has worked. In all situations, it helps if the typographical error in different weakly identifying fields is cleaned up.

The research problems are straightforward. Can multiple addresses improve matching? Do the multiple addresses need to be carried in two files A and B that are being matched? Can the multiple addresses be contained in an auxiliary population register? Are there other fields such as dates or names for which multiple versions might help matching? It is obvious that multiple addresses associated with an entity are not independent. How can matching weights (scores) be computed with multiple addresses? Business list matching is a major problem. If an agency maintains a large business register, how useful is it to maintain multiple versions of the name, addresses, and contact individuals associated with individual entities such as enterprises or company locations. In particular, do the multiple names and address improve the ability to detect duplicates within the list and to reduce the possible of a false duplicate (existing entity) being added from an external source?

4.13 **Maintaining Registers**

If a register is not maintained effectively, undetected duplication may increase by 1% or more per year. At the end of several years, it may be difficult to use the register for sampling, analysis and other purposes. The same is true for any large file contains a unique identifier such as a verified Social Security Number and is updated with source files that do not contain the unique identifier. Usually a register is the source of the unique identifier (such as a Social Security Number, Employer Identification Number or National Health Number). We begin by describing very straightforward issues and then proceed to more complicated issues.

An agency maintains a National Death Register (NDR). Entries are received from hospitals, coroner offices, and a variety of official sources. The first entry in Table 15 with source 'NDR' is the typical entry in the register after it has been cleaned up. The date-of-birth might be added from a record from a doctor, family member, coroner, or other source. We see that there are very minor variations in the name that either an automatic procedure or a clerically assisted procedure could easily deal with during matching. The county varies with some of the sources (police, doctor, family) and may represent counties that are near Harris County. Because the John Smith is a common name, the date-of-death with sources (coroner, police, doctor) help determine 'true' matches. The date-of-birth field is very useful in the NDR and might be available with records from the coroner or doctor (via health insurance verifying information). SSN can be useful for general purposes and might be added from an external source.

Table 15. Name and Other Information Variants

Source	Name	County	date-of-death	date-of-birth	SSN
NDR	John L. Smith	Harris County, TX	Jan 26, 2005	Jun 15, 1946	123-45-6789
Coroner	John Smith	Harris County, TX	Jan 26, 2005	Jun 15, 1946	bb
Police	John Smith	Next1 County, TX	Jan 26, 2005	bb	bb
Doctor	John L. Smith	Next1 County, TX	Jan 25, 2005	June 15, 1946	bb
Family	John L. Smith	Next2 County, TX	Feb 1, 2004	June 5, 1947	bb

We can make several observations. The NDR may have several sources from which it can update its register. In final entries, it may require extensive and periodic clerical review to remove typographical error. If NDR entries contain typographical error, then it is much more likely that duplicates from a source such as Doctor may be added erroneously. Over a period of time, individuals running the NDR may decide that certain sources such as Family may have much higher error rates and should only be used when other sources are unavailable. Any new entry may have typographical error that needs to be corrected. Any 'corrected' records should be certified with a status code that gives the date of the certification so that further changes to the record are not made. The SSN is a useful identifier because it allows better determination of whether an entry is a duplicate. There are 2-3 John Smiths associated with every date-of-birth and likely 2-3 John Smiths associated with most dates-of-death. It is possible that on some dates, two John Smiths die in Harris County, TX.

The research problems are straightforward. How much maintenance and resources are needed for a population register or large file C. How useful is it to clean up minor spelling variations and typographical error in files? Specifically, how are value-specific frequency tables that are constructed from population registers affected by typographical error? Do the errors affect less frequent strings more than less frequent strings?

5. **Concluding remarks**

This document provides an overview of record linkage. Record linkage is also referred to as data cleaning or object identification. It gives background on how record linkage has been applied in matching lists of businesses. It points out directions of research for improving the linkage methods.

References

Abowd, J. M. and Vilhuber, L. (2005), "The Sensitivity of Economic Statistics to Coding Errors in Personal Identifiers (with discussion)," *Journal of Business and Economic Statistics,* 23 (2), 133-165.

Abowd, J. M. and Woodcock, S. D. (2002), "Disclosure Limitation in Longitudinal Linked Data," in (P. Doyle et al, eds.) *Confidentiality, Disclosure, and Data Access,* North Holland: Amsterdam.

Abowd, J. M. and Woodcock, S. D. (2004), "Multiply-Imputing Confidential Characteristics and File Links in Longitudinal Linked Data, in (J. Domingo-Ferrer, and V. Torra, eds.), *Privacy in Statistical Databases 2004*, Springer: New York, 290-287.

Agichstein, E., and Ganti, V. (2004), "Mining Reference Tables for Automatic Text Segmentation," *ACM Knowledge Discovery and Data Mining Conference 2004*, 20-29.

Alvey, W. and Jamerson, B. (eds.) (1997), *Record Linkage Techniques -- 1997* (Proceedings of An International Record Linkage Workshop and Exposition, March 20-21, 1997, in Arlington VA), also published by National Academy Press (1999) and available at http://wwww.fcsm.gov under methodology reports.

Ananthakrishna, R., Chaudhuri, S., and Ganti, V. (2002), "Eliminating Fuzzy Duplicates in Data Warehouses," *Very Large Data Bases* 2002, 586-597.

Baxter, R., Christen, P., and Churches, T. (2003), "A Comparison of Fast Blocking Methods for Record Linkage," *Proceedings of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification,* Washington, DC, August 2003.

Belin, T. R., and Rubin, D. B. (1995), "A Method for Calibrating False- Match Rates in Record Linkage," *Journal of the American Statistical Association*, 90, 694-707.

Benjelloun, O., Garcia-Molina, H., Su, Q., and Widom, J. (2005), "Swoosh: A Generic Approach to Entity Resolution," Stanford University technical report, March 2005.

Bentley, J.L., and Sedgewick, R.A. (1960), "Fast Algorithms for Searching and Sorting Strings, Proceedings of the Eighth ACM-SIAM Symposium on Discrete Algorithms," 360-369.

Bertolazzi, P., De Santis, L., and Scannapieco, M. (2003), "Automatic Record Matching in Cooperative Information Systems," *Workshop on Data Quality in Cooperative Information Systems*, Siena, Italy, January, 2003.

Bilenko, M. and Mooney, R. J. (2003a), "Adaptive Duplicate Detection Using Learnable String Similarity Metrics," *Proceedings of ACM Conference on Knowledge Discovery and Data Mi*ning, Washington, DC, August 2003, 39-48.

Bilenko, M. and Mooney, R. J. (2003b), On Evaluation and Training-Set Construction for Duplicate Detection," *Proceedings of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification,* Washington DC, August 2003.

Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., and Fienberg, S. (2003), "Adaptive Name Matching in Information Integration," *IEEE Intelligent Systems*, 18 (50), 16-23.

Bertolazzi, P., De Santis, L., and M. Scannapieco, M. (2003) , "Automatic Record Matching in Cooperative Information Systems, " Proceedings of the Workshop on Data Quality in Cooperative Information Systems, Siena, Italy, January 2003.

Borkar, V., Deshmukh, K., and Sarawagi, S. (2001), "Automatic Segmentation of Text into Structured Records," Association of Computing Machinery SIGMOD 2001, 175-186.

Brefeld, U. and Scheffer, T. (2004), "Co-EM Support Vector Learning," *Proceedings of the 21st International Conference on Machine Learning*.

Bruni, R. (2004), "Discrete Models for Data Imputation," *Discrete Applied Mathematics*, 144, 59-69.

Bruni, R. (2005), "Error Correction for Massive Datasets," *Optimization Methods and Software*, 20 (2-3), 297-316.

Chaudhuri, S., Gamjam, K., Ganti, V., and Motwani, R. (2003), "Robust and Efficient Match for On-Line Data Cleaning," ACM SIGMOD 2003, 313-324

Chaudhuri, S., Ganti, V., and Motwani, R. (2005), "Robust Identification of Fuzzy Duplicates*," IEEE International Conference on Data Engineering*, 865-876.

Christen, P. Churches, T. and Zhu, J.X. (2002) "Probabilistic Name and Address Cleaning and Standardization," (The Australian Data Mining Workshop, November, 2002), available at http://datamining.anu.edu.au/projects/linkage.html .

Churches, T., Christen, P., Lu, J. and Zhu, J. X. (2002), "Preparation of Name and Address Data for Record Linkage Using Hidden Markov Models," *BioMed Central Medical Informatics and Decision Making*, 2 (9), available at http://www.biomedcentral.com/1472-6947/2/9/.

Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003a), "A Comparison of String Metrics for Matching Names and Addresses," *International Joint Conference on Artificial Intel*ligence, *Proceedings of the Workshop on Information Integration on the Web*, Acapulco, Mexico, August 2003.

Cohen, W. W., Ravikumar, P., and Fienberg, S. E. (2003b), "A Comparison of String Distance Metrics for Name-Matching Tasks," *Proceedings of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification,* Washington DC, August 2003.

Cohen, W. W. and Richman, J. (2002), "Learning to Match and Cluster Entity Names," ACM SIGKDD 2002, 475-480.

Cohen, W. W., and Sarawagi, S. (2004), "Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods," *Proceedings of the ACM Knowledge Discovery and Data Mining Conference 2005*, 89-98.

Cooper, W. S. and Maron, M. E. (1978), "Foundations of Probabilistic and Utility-Theoretic Indexing," *Journal of the Association of Computing Machinery*, 25, 67-80.

Culotta, A., and McCallum, A. (2005), "Joint Deduplication of Multiple Record Types in Relational Data," *CIKM* 2005.

Della Pietra, S., Della Pietra, V., and Lafferty, J. (1997), "Inducing Features of Random Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 380-393.

Deming, W. E., and Gleser, G. J. (1959), "On the Problem of Matching Lists by Samples," *Journal of the American Statistical Association*, 54, 403-415.

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.

Do, H.-H. and Rahm, E. "COMA – A system for flexible combination of schema matching approaches," Very Large Data Bases 2002, 610-621.

Dong, X., Halevy, A., and Madhavan, J. (2005), "Reference Reconciliation in Complex Information Spaces," *Proceedings of the ACM SIGMOD Conference* 2005, 85-96.

Elfekey, M., Vassilios, V., and Elmagarmid, A. "TAILOR: A Record Linkage Toolbox," *IEEE International Conference on Data Engineering 2002*, 17-28

Fan, W., Davidson, I., Zadrozny, B., and Yu, P. (2005), "An Improved Categorization of Classifier's Sensitivity on Sample Selection Bias, http://www.cs.albany.edu/~davidson/Publications/samplebias.pdf .

Fayad, U. and Piatetskey-Shipiro, G, and Smyth, P. (1996), "The KDD Process of Extracting Useful Information from Volumes of Data," *Communications of the Association of Computing Machinery*, 39 (11), 27-34.

Fayad, U. and Uthurusamy, R. (1996), "Data Mining and Knowledge Discovery in Data Bases," *Communications of the Association of Computing Machinery*, 39 (11), 24-26.

Fayad, U. and Uthurusamy, R. (2002), "Evolving Data Mining into Solutions for Insights," *Communications of the Association of Computing Machinery*, 45 (8), 28-31.

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

Ferragina, P. and Grossi, R. (1999), "The String B-tree: a New Data Structure for String Search in External Memory and Its Applications," *Journal of the Association of Computing Machinery*, 46 (2), 236-280.

Freund, Y. and Schapire, R. E. (1996), "Experiments with a New Boosting Algorithm," *Machine Learning: Proceedings of the Thirteenth International Conference*, 148-156.

Friedman, J., Hastie, T. ,Tibshirani, R. (2000), "Additive Logistic Regression: a Statistical View of Boosting," *Annals of Statistics*, 28, 337-407.

Gill, L. (1999), "OX-LINK: The Oxford Medical Record Linkage System," in *Record Linkage Techniques 1997*, Washington, DC: National Academy Press, 15-33.

Getoor, L., Friedman, N., Koller, D., and Taskar, B. (2003), "Learning Probabilistic Models for Link Structure," *Journal Machine Learning Research*, 3, 679-707.

Gravano, L., Ipeirotis, P. G., Jagadish, H. V., Koudas, N., Muthukrishnan, and Srivastava, D. (2001), "Approximate String Joins in a Database (Almost) for Free," *Proceedings of VLDB*, 491-500.

Guha, S., Koudas, N., Marathe, A., and Srivastava, D. (2004), "Merging the Results of Approximate Match Operations," *Proceedings of the 30th VLDB Conference*, 636-647.

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

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer: New York.

Hernandez, M. and Stolfo, S. (1995), "The Merge-Purge Problem for Large Databases," *Proceedings of ACM SIGMOD 1995*, 127-138.

Hjaltson, G. and Samet, H. (2003), "Index-Driven Similarity Search in Metric Spaces," *ACM Transactions On Database Systems*, 28 (4), 517-580.

Ishikawa, H. (2003), "Exact Optimization of Markov Random Fields with Convex Priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 1333-1336.

Iyengar, V. (2002), "Transforming Data to Satisfy Privacy Constraints," ACM KDD 2002, 279-288.

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.

Jin, L., Li, C., and Mehrortra, S. (2002), "Efficient String Similarity Joins in Large Data Sets," UCI technical Report, Feb. 2002, http://www.ics.uci.edu/~chenli/pub/strjoin.pdf .

Jin, L., Li, C., and Mehrortra, S. (2003), "Efficient Record Linkage in Large Data Sets," Eighth International Conference for Database Systems for Advance Applications (DASFAA 2003), 26-28

March, 2003, Kyoto, Japan, http://www.ics.uci.edu/~chenli/pub/dasfaa03.pdf .
Koller, D. and Pfeffer, A. (1998), "Probabilistic Frame-Based Systems," *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 580-587.
Koudas, N., Marathe, A., and Srivastava, D. (2004), "Flexible String Matching Against Large Databases in Practice," *Proceedings of the 30th VLDB Conference,*1078-1086.
Lafferty, J., McCallum, A., and Pereira, F. (2001), "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in Proceedings of the International Conference on Machine Learning, 282-289.
Lahiri, P., and Larsen, M. D. (2000), "Model-Based Analysis of Records Linked Using Mixture Models," American Statistical Association, *Proceedings of the Section on Survey Research Methods,* 11-19.
Lahiri, P. A. and Larsen, M. D. (2005 "Regression Analysis with Linked Data," *Journal of the American Statistical Association*, 100, 222-230.
Larsen, M. (1999), "Multiple Imputation Analysis of Records Linked Using Mixture Models," *Statistical Society of Canada, Proceedings of the Survey Methods Section*, 65-71.
Larsen, M. D. (2005), "Hierarchical Bayesian Record Linkage Theory," Iowa State University, Statistics Department Technical Report.
Larsen, M. D., and Rubin, D. B. (2001), "Alterative Automated Record Linkage Using Mixture Models," *Journal of the American Statistical Association*, 79, 32-41.
Lu, Q. and Getoor, L. (2003), "Link-based Classification," in (T. Fawcett and N. Mishra, eds.) *Proceedings of the Twentieth International Conference on Machine Learning*, 496-503.
Malin, B., Sweeney, L., and Newton, E. (2003), "Trail Re-Identification: Learning Who You Are From WhereYou Have Been," Workshop on Privacy in Data, Carnegie-Mellon University, March 2003.
McCallum, A., Bellare, K., and Pereira, F. (2005), "A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance," *UAI* 2005.
McCallum, A., Nigam, K., and Unger, L. H. (2000, "Efficient Clustering of High-Dimensional Data Sets with Application to Reference Matching, in *Knowledge Discovery and Data Mining*, 169-178.
McCallum, A. and Wellner, B. (2003), "Object Consolidation by Graph Partitioning with a Conditionally-Trained Distance Metric," *Proceedings of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification,* Washington DC, August 2003.
Michalowski, M., Thakkar, S., and Knoblock, C. A. (2003), "Exploiting Secondary Sources for Object Consolidation," *Proceedings of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification,* Washington DC, August 2003.
Mitchell, T. M. (1997), *Machine Learning*, New York, NY: McGraw-Hill.
Navarro, G. (2001), "A Guided Tour of Approximate String Matching," *Association of Computing Machinery Computing Surveys*, 33, 31-88.
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.
Newcombe, H.B. and Kennedy, J. M. (1962) "Record Linkage: Making Maximum Use of the Discriminating Power of Identifying Information" *Communications of the Association for Computing Machinery*, 5, 563-567.
Newcombe, H. B. and Smith, M. E. (1975), "Methods for Computer Linkage of Hospital Admission-Separation Records into Cumulative Health Histories," *Methods of Information in Medicine*, 14 (3), 118-125.
Ng, A. and Jordan, M. (2002), "On Discriminative vs. Generative classifiers: A comparison of logistic regression and naïve Bayes," *Neural Information Processing Systems 14.*
Pasula, H., Baskara, M., Milch, B., Russell, S., and Shipster, I. (2003), "Identity Uncertainty and Citation Matching," *NIPS* 03.
Pasula, H. and Russell, S. (2001), "Approximate Inference for First-Order Probabilistic Languages," *Proceedings of the International Joint Conference on Artificial Intelligence*, 741-748.
Pasula, H., Russell, S., Ostland, M., and Ritov, Y. (1999), "Tracking Many Objects with Many Sensors," *Proceedings of the Joint International Conference on Artificial Intelligence*.
Pollock, J. and Zamora, A. (1984), "Automatic Spelling Correction in Scientific and Scholarly Text," *Communications of the ACM*, 27, 358-368.
Porter, E. H., and Winkler, W. E. (1999), "Approximate String Comparison and its Effect in an Advanced

Record Linkage System," in Alvey and Jamerson (ed.) *Record Linkage Techniques - 1997*, 190-199, National Research Council, National Academy Press: Washington, D.C.
Rahm, E. and Do, H.-H. (2000), "Data Cleaning: Problems and Currrent Approaches," *IEEE Bulletin on Data Engineering*, 23 (4).
Ravikumar, P. and Cohen, W. W. (2004), "A Hierarchical Graphical Model for Record Linkage," *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Banff, Calgary, CA, July 2004.
Ravikumar, P. and Lafferty, J. (2004), "Variational Chernoff Bounds for Graphical Models," *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, Banff, Calgary, CA, July 2004
Ristad, E. S., and Yianilos, P. (1998), "Learning String-Edit Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 522-531.
Russell, S. (2001), "Identity Uncertainty," *Proceedings of IFSA-01*, http:/www.cs.berkeley.edu/~russell/papers/ifsai01-identity.ps .
Sarawagi, S. and Bhamidipaty, A. (2002), "Interactive Deduplication Using Active Learning, *Very Large Data Bases 2002, 269-278*.
Scannipieco, M. (2003), "DAQUINCIS: Exchanging and Improving Data Quality in Cooperative Information Systems," Ph.D. thesis in Computer Engineering, University of Rome "La Sapienza."
Scheuren, F., and Winkler, W. E. (1993), "Regression analysis of data files that are computer matched," *Survey Methodology*, 19, 39-58.
Scheuren, F., and Winkler, W. E. (1997), "Regression analysis of data files that are computer matched, II," *Survey Methodology*, 23, 157-165.
Sebastiani, F. (2002), "Machine Learning in Automated Text Categorization," *Association of Computing Machinery Computing Surveys*, 34 (1), 1-47.
Sekar, C. C., and Deming, W. E. (1949), "On a Method of Estimating Birth and Death Rates and the Extent of Registration," *Journal of the American Statistical Association*, 44, 101-115.
Steel, P., and Konshnik, C. (1999), "Post-Matching Adminstrative Record Linkage Between Sole Proprietorship Tax Returns and the Standard Statistical Establishment List," in *Record Linkage Techniques 1997*, Washington, DC: National Academy Press, 179-189.
Sweeney, L. (1999), "Computational Disclosure Control for Medical Microdata: The Datafly System" in *Record Linkage Techniques 1997*, Washington, DC: National Academy Press, 442-453.
Taskar, B., Abdeel, P., and Koller, D. (2002), "Discriminative Probabilistic Models for Relational Data," *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.
Taskar, B., Segal, E., and Koller, D. (2001), Probabilistic Classification and Clustering in Relational Data," *Proceedings of the International Joint Conference on Artificial Intelligence.*
Taskar, B., Wong, M. F., Abdeel, P. and Koller, D. (2003), "Link Prediction in Relational Data," *Neural Information Processing Systems*, to appear.
Taskar, B., Wong, M. F., and Koller, D. (2003), "Learning on Test Data: Leveraging "Unseen" Features," *Proceedings of the Twentieth International Conference on Machine Learning*, 744-751.
Tejada, S., Knoblock, C., and Minton, S. (2001), "Learning Object Identification Rules for Information Extraction," *Information Systems*, 26 (8), 607-633.
Tejada, S., Knoblock, C., and Minton, S. (2002), "Learning Domain-Independent String Transformation for High Accuracy Object Identification, *Proc. ACM SIGKDD '02.*
Torra, V. (2004), "OWA Operators in Data Modeling and Re-Identification," *IEEE Transactions on Fuzzy Systems*, 12 (5) 652-660.
Vapnik, V. (2000), *The Nature of Statistical Learning Theory (2nd Edition)*, Berlin: Springer-Verlag.
Wang, S., Schuurmans, D., Peng, F., and Zhao, Y. (2003), "Learning Mixture Models with the Latent Maximum Entropy Principal," in (T. Fawcett and N. Mishra, eds.) *Proceedings of the Twentieth International Conference on Machine Learning*, 776-783 (also version for *IEEE Transactions on Neural Nets* in 2004).
Wei, J. (2004), Markov Edit Distance, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 26 (3), 311-321.
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. (1989a), "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. (1989b), "Methods for Adjusting for Lack of Independence in an Application of the

Fellegi-Sunter Model of Record Linkage," *Survey Methodology*, 15, 101-117.
Winkler, W. E. (1989c), "Frequency-based Matching in the Fellegi-Sunter Model of Record Linkage," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, 778-783.
Winkler, W. E. (1990a), "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. (1990b), "On Dykstra's Iterative Fitting Procedure," *Annals of Probability*, 18, 1410-1415.
Winkler, W. E. (1993a) "Business Name Parsing and Standardization Software," unpublished report, Washington, DC: Statistical Research Division, U.S. Bureau of the Census.
Winkler, W. E. (1993b), "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. (1994), "Advanced Methods for Record Linkage," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, 467-472 (longer version report 94/05 available at http://www.census.gov/srd/www/byyear.html).
Winkler, W. E. (1995), "Matching and Record Linkage," in B. G. Cox *et al*. (ed.) *Business Survey Methods*, New York: J. Wiley, 355-384 (also available at http://www.fcsm.gov/working-papers/wwinkler.pdf).
Winkler, W. E. (1998). "Re-identification Methods for Evaluating the Confidentiality of Analytically Valid Microdata," *Research in Official Statistics*, 1, 87-104.
Winkler, W. E. (1999a). "The State of Record Linkage and Current Research Problems," *Statistical Society of Canada, Proceedings of the Survey Methods Section*, 73-80 (longer version at http://www.census.gov/srd/www/byyear.html).
Winkler, W. E. (1999b), "Issues with Linking Files and Performing Analyses on the Merged Files," *Proceedings of the Sections on Government Statistics and Social Statistics*, *American Statistical Association*, 262-265.
Winkler, W. E. (1999c), "Record Linkage Software and Methods for Administrative Lists," Eurostat, *Proceedings of the Exchange of Technology and Know-How '99,* also available at http://www.census.gov/srd/www/byyear.html.
Winkler, W. E. (2000), "Machine Learning, Information Retrieval, and Record Linkage," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, 20-29. (also available at http://www.niss.org/affiliates/dqworkshop/papers/winkler.pdf).
Winkler, W. E. (2001), "The Quality of Very Large Databases," *Proceedings of Quality in Official Statistics '2001*, CD-ROM (also available at http://www.census.gov/srd/www/byyear.html as report rr01/04).
Winkler, W. E. (2002), "Record Linkage and Bayesian Networks," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, CD-ROM (also at http://www.census.gov/srd/www/byyear.html).
Winkler, W. E. (2003a), "Methods for Evaluating and Creating Data Quality," *Proceedings of the Workshop on Cooperative Information Systems*, Sienna, Italy, January 2003, longer version in *Information Systems* (2004), 29 (7), 531-550.
Winkler, W. E. (2003b), "Data Cleaning Methods," *Proceedings of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification*, Washington, DC, August 2003.
Winkler, W.E. (2004a), "Re-identification Methods for Masked Microdata," in (J. Domingo-Ferrer and V. Torra, eds.) *Privacy in Statistical Databases 2004*, New York: Springer, 216-230, http://www.census.gov/srd/papers/pdf/rrs2004-03.pdf .
Winkler, W.E. (2004b), "Masking and Re-identification Methods for Public-Use Microdata: Overview and Research Problems," in (J. Domingo-Ferrer and V. Torra, eds.) *Privacy in Statistical Databases 2004*, New York: Springer, 231-247, http://www.census.gov/srd/papers/pdf/rrs2004-06.pdf .
Winkler, W. E. (2004c), "Approximate String Comparator Search Strategies for Very Large Administrative Lists," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, CD-ROM (also available as report 2005/02 at http://www.census.gov/srd/www/byyear.html).
Winkler, W. E. (2006), "Sample Allocation and Stratification," in (P.S.R.S. Rao and M. Katzoff) *Handbook on Sampling Techniques and Analysis*, CRC Press: London, UK.

Winker, W. E., and Yancey, W. E. (2006), "Record Linkage Error-Rate Estimation without Training Data," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, to appear.

Winker, W. E., and Yancey, W. E. (2006), "Parallel BigMatch," technical report, to appear.

Yancey, W.E. (2000), "Frequency-Dependent Probability Measures for Record Linkage," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, 752-757 (also at http://www.census.gov/srd/www/byyear.html).

Yancey, W.E. (2002), "Improving EM Parameter Estimates for Record Linkage Parameters," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, CD-ROM (also at http://www.census.gov/srd/www/byyear.html as report RRS 2004/01).

Yancey, W.E. (2003), "An Adaptive String Comparator for Record Linkage," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, CD-ROM (also report RRS 2004/02 at http://www.census.gov/srd/www/byyear.html).

Yancey, W.E. (2004), "The BigMatch Program for Record Linkage," *Proceedings of the Section on Survey Research Methods*, *American Statistical Association*, CD-ROM.

Yancey, W.E. (2005), "Evaluating String Comparator Performance for Record Linkage," research report RRS 2005/05 at http://www.census.gov/srd/www/byyear.html).

Yancey, W.E., and Winkler, W. E. (2004), "BigMatch Software," computer system, documentation available at http://www.census.gov/srd/www/byyear.html as report RRC 2002/01).

Yancey, W.E., Winkler, W.E., and Creecy, R. H. (2002) "Disclosure Risk Assessment in Perturbative Microdata Protection," in (J. Domingo-Ferrer, ed.) *Inference Control in Statistical Databases*, Springer: New York.

Zadrozny, B. (2004), "Learning and Evaluating Classifiers under Sample Selection Bias," *Proceedings of the International Conference on Machine Learning*, .

Zhang, T. and Oles, F. (2001), "Text Categorization Based on Regularized Linear Classification Methods," *Information Retrieval*, 4 (1), 5-31.