Data Cleaning Methods

William E Winkler 1/
U.S. Bureau of the Census
Statistical Research, Room 3000-4
Washington, DC 20233-9100
001-301-763-4729
william.e.winkler@census.gov

ABSTRACT

Data Cleaning methods are used for finding duplicates within a file or across sets of files. This overview provides background on the Fellegi-Sunter model of record linkage. The Fellegi-Sunter model provides an optimal theoretical classification rule. Fellegi and Sunter introduced methods for automatically estimating optimal parameters without training data that we extend to many real world situations.

Keywords

EM Algorithm, string comparator, unsupervised learning.

1. INTRODUCTION

Methods for finding duplicates are referred to as data cleaning, object identification, or record linkage. This paper provides an overview of the Fellegi-Sunter method of record linkage and various ways of implementing it. Fellegi and Sunter [10] provided a mathematical model for record linkage that proved the optimality of a linkage (classification) rule introduced by Newcombe [19,20]. They introduced methods for automatically estimating optimal parameters without training data.

To prepare for various extensions of the methods, we provide the notation and describe the main theorem of Fellegi and Sunter. We begin by describing the main methods of unsupervised learning and their extensions to many real-world situations. The extensions involve dependencies between fields, determining suitable sets of pairs rather than using all pairs from two files, and, in a very narrow set of situations, automatic estimation of error rates. More recent research has involved using auxiliary files to improve linkages when insufficient information is available within the two files being matched. We describe methods of analytic linking that create new information during the matching process to improve both the matching and resultant statistical analyses on sets of linked files. We also describe the BigMatch technology for a matching moderate size file of 100 million records against large administrative lists having upwards of 4 billion records. BigMatch technology can significantly reduce disk storage requirements (75%), cpu time (75%), and skilled programmer intervention (90%) for large projects.

Conference SIGKDD '03, August 24-27, 2003, Washington, DC.

2. BACKGROUND

In this section, we describe the main model of record linkage, automatic methods of parameter estimation and error-rate estimation, methods of accounting for dependencies between fields, methods of finding better sets of pairs on which matching is done, and some straightforward extensions to situations when small amounts of labeled training data can be combined with unlabeled data.

2.1 Fellegi-Sunter Model of Record Linkage

Fellegi and Sunter [10] provided a formal mathematical model for ideas that had been introduced by Newcombe [19,20]. They provided many ways of estimating key parameters. To begin, notation is needed. Two files **A** and **B** are matched. The idea is to classify pairs in a product space $\mathbf{A} \times \mathbf{B}$ from two files A and B into M, the set of true matches, and U, the set of true nonmatches. Fellegi and Sunter 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).

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 the rates of 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 γ

$\in \Gamma$ 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.

Pairs with weights above the upper cut-off are referred to as *designated matches* (or links). Pairs below the lower cut-off are referred to as *designated nonmatches* (or nonlinks). The remaining pairs are referred to as *designated potential matches* (or potential links). If $T_\mu = T_\lambda$, then decision rule (1) can be used for separating records (correspondingly pairs) into those that are in one class from those that are not. The probabilities P(agree first | M), P(agree last | M), P(agree age | M), P(agree first | U), P(agree last | U), and P(agree age | U) are called *marginal probabilities*. P(| M) & P(| U) are called the m- and u-probabilities, respectively. The logarithms of the ratios of probabilities associated with individual fields (marginal probabilities) are called the *individual agreement weights*. The m- and u-probabilities are also referred to as *matching parameters*. A *false match* is a pair that is designated as a match and is truly a nonmatch. A *false nonmatch* is pair that is designated as a nonmatch and is a truly a match.

2.2 Automatic Parameter Estimation without Training Data

Fellegi and Sunter [10] 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 [34]. Winkler [27] showed how to estimate the probabilities using the EM-Algorithm [8,16]. Winkler [28] demonstrated that the EM algorithm estimates optimal parameters in some situations. The estimated probabilities are particularly useful because the optimal m- and u-probabilities can vary significantly from one region of the U.S. to another. In particular, the conditional probability P(agreement on first name | M) can differ significantly from an urban region to an adjacent suburban region [28].

Belin and Rubin [1] introduced methods of automatic-error rate estimation that used information from the basic matching situations of Winkler [28]. Their error rate estimates were reasonably accurate that Scheuren and Winkler [24] could use the estimated error rates in a statistical model that adjusts regression analyses for linkage errors. Winkler [33] observed, however, that the 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 problems. With many administrative lists, business lists, and agriculture lists, the methods of Belin and Rubin were unsuitable. The general problem of error rate estimation is very difficult. It is known as the regression problem [12].

2.3. Suitable Subsets of Pairs from A × B

Although the EM algorithm is a method of unsupervised learning [18,31] that will not generally find two classes C_1 and C_2 that correspond to M and U, Winkler [28] demonstrated that it does well in a few suitable situations. In those situations, the classes C_1 and C_2 that correspond closely to M and U and the parameters of the form $P(\gamma|M)$ and $P(\gamma|U)$ can be nearly optimal. In most matching situations, we cannot consider all pairs from two files A and B. We usually consider pairs that agree on a geographic identifier such as the U.S. Postal ZIP code (if available) and an additional characteristic such as first character of surname. In record linkage, matching on subsets of pairs agreeing on a set of characteristics is called *blocking*. The best situation in when a set of blocking variables yields a set of pairs that contains a proportion of matches $P(\gamma)$ that is above 0.03. For instance, blocking on the combination of a ZIP+4 code and first character of surname can yield sets of pairs having match proportion $P(\gamma)$ between 0.01 and 0.1. If first name, surname, age, and sex are available, then the EM algorithm yields very good parameters for most situations.

If we consider all pairs agreeing on only on the combination of ZIP code and first character of surname, then the proportion of pairs $P(\gamma)$ that are matches can drop to 0.0001. Any unsupervised learning method will not be able parameters that separate the pairs into two classes that approximate M and U. To alleviate the situation, we take an initial guess of the parameters $P(\gamma)$, and $P(\gamma|M)$ and $P(\gamma|U)$. Within the set of pairs agreeing on the combination of ZIP code and first character of surname, we only consider those pairs having likelihood ratio (1) above a certain point. In virtually all situations, this will yield a set of pairs in which the proportion of matches $P(\gamma)$ is above 0.03 and for which the EM can yield optimal parameters. Recent verification of the efficacy of this approach and extensions to the methods for choosing the pairs are due to Yancey [36] and Elfekey et al. [9].

2.4 Accounting for Dependencies between Fields

Individual fields used in matching can have strong dependencies between them. For instance, assume that we are using name, address, sex, and age to match household data. Assume that we are only considering pairs that are brought together using ZIP+4 and first character of the surname. If a pair of records agrees on last name, then the pair is likely, with probability close to one, to agree of household characteristics such as house number, street name, and phone number [30,31]. Pairs of records representing two individuals from the same household can be associated with both matches and nonmatches. If ratio (1) is computed under the conditional independence assumption (i.e., naïve Bayes), then the accuracy of decision rules (2) may be reduced in comparison to situations in which dependencies between variables are accounted for. One solution is to continue use of the conditional independence EM but to use three classes. In the same household, the variables first name, sex, and age are used to separate matches from nonmatches. This has the effect of subdividing pairs in the same household into matches and nonmatches. The resultant parameters can be combined into those associated with two classes when decision rule (2) is applied. This partially accounts for dependencies of the household variables.

Alternatively, we can use general EM methods [30,17] that account for interactions of the variables. In the M-step, a general iterative fitting algorithm [30] that generalizes the iterative scaling algorithm of Della Pietra et al. [7] is used. The theoretical and computational aspects are fully described in [30]. In particular, convex constraints (possibly based on prior knowledge) can be used to predispose the parameters $P(\gamma|M)$ and $P(\gamma|U)$ to subsets of the parameter space. In a narrow range of situations, the parameters obtained by the general fitting procedures yield both good decision rules and accurate estimates of the error rates [30]. The accuracy of the estimates of error rates is partially confirmed in Larsen and Rubin [14] who extended the EM methods with general MCMC methods.

2.5 Combining Labeled Training Data with Unlabeled Data

In machine learning, it is quite typical to assume that training data will be available. The training is used for getting parameters needed for the classification (decision) rules of the method being applied. In record linkage, because training data is seldom available, unsupervised learning under conditional independence is typically used. The theoretical and computational methods of Winkler [30] extend to situations where a combination of labeled training data and unlabeled data are used. The reason for using labeled training data is that unsupervised learning methods will not always give reasonable estimates of parameters $P(\gamma|M)$. For instance, typographical error rates can vary significantly between an urban region and an adjacent suburban. The probability on age and first name given a match, in particular, can be much lower in the urban area than in the suburban area.

Because it is expensive to obtain labeled training data, individuals have used methods for combining a small amount of training data with a large amount of unlabelled data. Nigam et al. [21] showed how to do the combining in a text classification application in applying naïve Bayes networks. Winkler [34] extended the use of labeled and unlabeled data to a model where various interactions could be accounted for. In a record linkage application, Winkler [35] demonstrated how the use of small amounts of training data could be combined with unlabeled data in a record linkage application. The main advantage of the record linkage application was that it yielded reasonably accurate estimates of error rates for a large class of situations than could not be done by the Belin and Rubin [1] methods. A secondary advantage was that it can be used in datamining experiments to determine what are the suitable interactions between the matching fields. Winkler [35] further observed that a sufficient amount of training data was needed for combining with the unlabeled data. The training data had a tendency to significantly reduce the number of computational paths and get the estimates closer to those obtained with large amounts of training data. In earlier work, Winkler [30] observed that purely unsupervised methods would often yield parameter estimates that were totally unsuitable for accurate decision rules. The unsuitable estimates even occurred in situations where convex constraints were used to predispose the estimated parameters to various subregions of the parameter space.

3. ADVANCED METHODS

In this section we consider advanced methods for bridging files and matching exceptionally large files via BigMatch technology.

3.1 Bridging Files

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, 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, 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 1. Linking Inputs and Outputs from Companies

Economics- Companies

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

Table 2. 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 is where name, address, and other information is 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 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, 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, then we might further reduce the number of records in the B-file that record r_0 can be linked to. The reduction might be to one or zero records in the B file that record r_0 can be linked to.

Each of the linkages and reductions in the number of B-file records that r_0 can be linked to 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.

Table 3. 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}

3.2 BigMatch Technology

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 via use 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, and year-of-birth have 5% typographical error rates, then exact character-by-character matching across quarters could miss upwards of 25% of matches.

BigMatch technology alleviates the limitations of the classical matching situation [37]. Only the smaller B file and appropriate indexes are held in core. 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 basic quick sort method of Bentley and Sedgewick [2]. In testing, we found the Bentley and Sedgewick sort to be faster than three other quick sort algorithms. The second index gives a very fast method of retrieving a comparing the records information from the B-file with individual records from the A-file. A B-file of 100 million records can reside in 4 gigabytes of memory. Only one pass is made on the B-file. Several output streams are created for each 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. The BigMatch matching software is nearly as fast as a classical matching program that makes multiple passes. 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. The biggest savings is often from the reduction in the amount of skilled intervention from the individuals running the software.

4. RELATED METHODS

There are three closely related areas of research involving (1) preprocessing and standardization methods for identifying subfields and making them more comparable, (2) advanced string comparators and their effect on the classification rules and (3) analytic linking methods for creating extra information during the linkage process to improve linkage and resultant analyses of linked files.

4.1 Preprocessing

Winkler [32] describes methods of preprocessing for business names and general addresses. For a name, we wish to replace titles such as ‘Doctor’ and ‘Dr.’ and words such as ‘Corporation’ and ‘Corp.’ with standard spellings such as ‘DR’ and ‘CORP’, respectively. When applicable, we wish to identify words such as first name, middle name, and last name so that they can be compared. We also wish do standardization on addresses. Winkler [32] applies rule-based logic in a business name standardizer that also works well agriculture lists having partnerships and with person lists. He also applies rule-based methods developed by the Census Geography Division for standardizing addresses.

Borkar et al. [3] introduced hidden Markov models for address standardization. The methods require training data and are adaptable to Asian-types of addresses that are particularly difficult for the rule-based methods. Christen et al. [5] and Churches et al. [6] apply hidden Markov models to both names and addresses. They show that it is quite straightforward to quickly generate training data by clerically standardizing a small number of records. They confirm that the hidden Markov methods work well for addresses. In this initial application of hidden Markov models, the methods work poorly for names [6] when compared to rule-based methods.

4.2 Approximate String Comparison

Typographical variations in the spelling of words are prevalent in computer files. Winkler [31] observed that approximately 25% of first names and 15% of last names for matches could not be compared character-by-character in three sites of a dress rehearsal Census. Using ideas of Pollock and Zamora [22], he developed a variant of the Jaro string comparator. He modeled methods for how various levels of partial agreement in the string comparator affect the likelihood ratio (1). Because the string comparators perform favorably in comparison with Bigrams and Edit-Distance, it has been adopted in the commercial GRLS system of Statistics Canada and Choicemaker software [4]. Sarawagi and Bhamidipaty [23] recently applied the new string comparator.

4.3 Analytic Linking

Datamining groups of files is intended to allow analyses that were not previously possible with single files. The linkages may also increase the accuracy of analyses. In some situations, it may be possible to create more information to improve the linkage process and to determine how much an analysis such as a regression can be improved using a theoretical model of the linkage error [24,25].

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 3). 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 [25] 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

Other methods also create information during the matching process that improves the identifications of duplicates. Probabilistic Relational Models [13,11] use sophisticated methods of clustering on different variables to improve the matching using naïve Bayesian networks. Malin et al. [15] use Reidentification of Data in Trails (REIDIT) algorithms for tracking and identifying individuals visiting a set of web pages. The methods might be used to extend the bridging-file ideas. Torra [26] introduced ordered weighted aggregation (OWA) operators to create information for linkages when there are no common identifying variables.

4. CONCLUDING REMARKS

This document provides an overview of a number of methods for finding duplicates within and across files. Some of the most advanced methods use statistical ideas that correspond to and generalize methods from the computer science literature. One novelty is the emphasis on sophisticated methods of unsupervised learning for getting optimal matching parameters. Another is BigMatch technology for efficiently matching moderate size files having a 100 million or more records.

1/ This paper reports the results of research and analysis undertaken by Census Bureau staff. It has undergone a Census Bureau review more limited in scope than that given to official Census Bureau publications. This report is released to inform interested parties of research and to encourage discussion.

5. REFERENCES

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

[2] Bentley, J.L., and Sedgewick, R.A. Fast Algorithms for Searching and Sorting Strings, Proceedings of the Eighth ACM-SIAM Symposium on Discrete Algorithms, (1996), 360-369.

[3] Borkar, V., Deshmukh, K., and Sarawagi, S. Automatic Segmentation of Text into Structured Records, Association of Computing Machinery SIGMOD '01, (2001).

[4] Borthwick, A. MEDD 2.0, (Conference Presentation, New York City, NY, USA, February, 2002), available at http://www.choicemaker.com.

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

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

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

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

[9] Elfekey, M., Vassilios, V., and Elmagarmid, A. TAILOR: A Record Linkage Toolbox, IEEE International Conference on Data Engineering '02, (2002).

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

[11] Getoor, L., Friedman, N., Koller, D., and Taskar, B. Learning Probabilistic Models of Relational Structure, ICML '01, (2001).

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

[13] Koller, D., and Pfeffer, A. Probabilistic Frame-Based Systems, Proc. AAAI, (1998).

[14] Larsen, M. D., and Rubin, D. B. AIterative Automated Record Linkage Using Mixture Models,Journal of the American Statistical Association, 79, (2001), 32-41.

[15] Malin, B., Sweeney, L., and Newton, E. Trail Re-Identification: Learning Who You are from Where You Have Been, presented at the Workshop on Privacy in Data, (Carnegie-Mellon University, March 2003).

[16] McLachlan, G. J., and Krisnan, T. The EM Algorithm and Extensions, John Wiley: New York, (1997).

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

[18] Mitchell, T. M. Machine Learning, New York, NY: McGraw-Hill, (1997).

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

[20] Newcombe, H.B., and Kennedy, J. M. Record Linkage: Making Maximum Use of the Discriminating Power of Identifying Information. Communications of the Association for Computing Machinery, 5, (1962) 563-567.

[21] Nigam, K., McCallum, A. K., Thrun, S., and Mitchell, T. Text Classification from Labeled and Unlabeled Documents using EM, Machine Learning, 39, (2000), 103-134.

[22] Pollock, J., and Zamora, A. Automatic Spelling Correction in Scientific and Scholarly Text, Communications of the ACM, 27, (1984), 358-368.

[23] Sarawagi, S., and Bhamidipaty, A. Interactive Deduplication Using Active Learning, Very Large Data Bases '02, (2002).

[24] Scheuren, F., and Winkler, W. E. Regression analysis of data files that are computer matched, Survey Methodology, 19, (1993), 39-58.

[25] Scheuren, F., and Winkler , W. E. Regression analysis of data files that are computer matched, II, Survey Methodology, 23, (1997), 157-165.

[26] Torra, V. Re-Identifying Individuals Using OWA Operators, Proceedings of the 6h International Conference on Soft Computing (Iizuka, Fukuoka, Japan, 2000).

[27] Winkler, W. E. 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, (1988), 667-671.

[28] Winkler, W. E. Near Automatic Weight Computation in the Fellegi-Sunter Model of Record Linkage, Proceedings of the Fifth Census Bureau Annual Research Conference, (1989), 145-155.

[29] Winkler, W. E. On Dykstra's Iterative Fitting Procedure, Annals of Probability, 18, (1990), 1410-1415.

[30] Winkler, W. E. Improved Decision Rules in the Fellegi-Sunter Model of Record Linkage, Proceedings of the Section on Survey Research Methods, American Statistical Association, (1993), 274-279.

[31] Winkler, W. E. Advanced Methods for Record Linkage, Proceedings of the Section on Survey Research Methods, American Statistical Association, (1994), 467-472 (longer version report 94/05 available at http://www.census.gov/srd/www/byyear.html).

[32] Winkler, W. E. Matching and Record Linkage, in B. G. Cox et al. (ed.) Business Survey Methods, New York: J. Wiley, (1995), 355-384.

[33] Winkler, W. E. The State of Record Linkage and Current Research Problems, Statistical Society of Canada, Proceedings of the Section on Survey Methods, (1999), 73-79 (longer version report rr99/04 available at http://www.census.gov/srd/www/byyear.html).

[34] Winkler, W. E. Machine Learning, Information Retrieval, and Record Linkage, Proceedings of the Section on Survey Research Methods, American Statistical Association, (2000), 20-29. (available at http://www.niss.org/affiliates/dqworkshop/papers/winkler.pdf).

[35] Winkler, W. E. Record Linkage and Bayesian Networks, Proceedings of the Section on Survey Research Methods, American Statistical Association, (2002), to appear (also at http://www.census.gov/srd/www/byyear.html).

[36] Yancey, W. E. Improving EM Algorithm Estimates for Record Linkage Parameters, Proceedings of the Section on Survey Research Methods, American Statistical Association, (2002), to appear.

[37] Yancey, W. E., and Winkler, W. E. BigMatch software, computer system, (2003), documentation of first version is in research report RRC2002/01 at http://www.census.gov/srd/www/byyear.html .