Application of Automated Records Linkage Software in Traffic Records Analysis

KARL KIM AND LAWRENCE NITZ

Following a brief discussion of the underlying theory of records linkage, an automated record linkage software package called Automatch is examined along with its various applications. Features, hardware, and user requirements are discussed, and user support and interfaces are detailed and commented on. An example linking crash reports to ambulance records is described. After other possible applications and uses for this software are described, additional issues about records linkage are raised. The report is part of ongoing research carried out by the Hawaii Crash Outcome Data Evaluation System (CODES) project, funded by the National Highway Safety Traffic Administration, U.S. Department of Transportation. The purpose of the CODES Project is to link crash, EMS, hospital, claims, and long-term care data to conduct analyses on the effectiveness of seat belts, motorcycle helmets, and other traffic safety interventions.

At one level, linking records from different data bases raises interesting and complex questions about privacy and the uses of computerized data. When one considers the many data bases that have been created and adds the possibility of linkages among these data bases, frightening, Orwellian images could be invoked. At a second level, there are other questions involving how to accomplish such a task. Recent developments in the theory and methods of records linkage, including the release of a software package called Automatch (*1*), serve to enhance the feasibility of records linkage. These technological developments may in turn spark more debate and discussion on issues of the uses of data and appropriateness of linking diverse data bases together. This report deals principally with the second-order concerns—that is, how to use available methods and technology to carry out records linkage. While outside the scope of this paper, the basic concerns about the appropriateness and ethics of data linkage must also be addressed. Recent advances in technology point to some areas of concern that are summarized in the conclusions.

Perhaps every social science researcher has at one point linked or tried to link two different data bases. Typically, the data bases were collected by different agencies for different purposes, but they provide valuable information. Studies have linked land-use data with tax data, typically at the parcel level. Transportation data (car ownership, drivers licenses, etc.) could also be linked to housing data bases and to zoning data bases. School enrollment figures are often pooled with other data bases to derive estimates of population change. Other social data such as health statistics, crime surveys, and many different surveys and opinion polls are often used for purposes for which they might not have been initially designed. This is the nature of data collection—the cardinal rule is often to use existing data bases before expending the time and resources to gather what would amount to essentially the same information.

Department of Urban Planning, University of Hawaii, Porteus 107 2424 Maile Way, Honolulu, Hawaii 96822.

More data are becoming available in computerized form so that it is not at all unusual to pass machine-readable data (tape, diskette, or CD-ROM) between different users. At the same time, more users are becoming computer literate with the proliferation of PCs, workstations, and statistical packages. Yet the merging of different data bases still poses some basic difficulties. First, surveys and other data bases generally protect anonymity so that unique identifiers such as name or social security number and so forth are not used. Second, even if name, street address, or other identifiers are available, there are still problems with matching records because of inconsistencies across sources in data entry and editing procedures. For example, the use of initials instead of full names, different abbreviations for street names, and the usual assortment of misspellings and other errors in the data base make exact matches impossible.

Several years ago, a survey on attitudes toward helmet laws in Hawaii was conducted. To construct a sample, two different data bases were linked: the vehicle registration file and the operator's license file. For the city and county of Honolulu in 1989, there were 8,514 registered motorcycles. (Military Personnel were excluded from the population.) For the same year, 13,595 persons held motorcycle licenses. It is not expected that everyone who has a motorcycle license owns a motorcycle and vice versa. Yet in terms of producing the best sample of motorcyclists, it appeared reasonable to construct a single file consisting of those who both were licensed and owned motorcycles. Because of misspellings, differences in punctuation, and other differences in the information contained in the two files, few records could be exactly matched. A matching strategy was devised to organize records in both files around the name field, then to match on the basis of last name, first name, street address, and zip code using the Statistical Analysis System (SAS) statistical package. Once the exact matches were located by computer, all remaining pairs were reviewed manually. On the basis of name and address, only 2,970 cases were matched, less than 35 percent of the registered motorcycles.

Manual review took many hours and, serious problems are associated with this procedure. Certain people, particularly those who tended to move or change addresses frequently, were more likely to be excluded, which could introduce certain biases. Some individuals in the ownership file owned more than one motorcycle and therefore showed up as duplicates in the ownership file but as unique records in the license file. Finally, uncontrolled error was introduced by the manual review process—in addition to being tiresome work, the process of comparing records to identify a match is tedious, particularly because it is difficult to devise a comprehensive set of decision rules without reviewing all the data. Each of these problems, from the duplicate records in the registration file to the problems associated with manual comparison of records, could have been handled more effectively with the Automatch software.

THEORY OF RECORD LINKAGE

Although a detailed mathematical discussion on the theoretical developments of record linkage is outside the purview of this paper, it is important to note that there have been many important theoretical developments. The Automatch software builds on the work of Felligi and Sunter (*2*), Newcombe and Kennedy (*3*), Newcombe (*4*), Jaro (*5*), and others. Although there are a few commercially available programs for records linkage, the Automatch program is experiencing growing popularity—not just among those interested in records linkage but also among those involved in more specialized activities of geocoding and data base management. The geocoding and record unduplicating features of the program will be discussed later.

To conceptualize the theory behind the Automatch software, one must begin with two different files. Each file contains fields of fixed length and a finite number of records. For records to match on these two different files they must share one or more equivalent fields. If every common field contributes to linking, the larger the number of common fields in the two files, the greater the opportunities for linking the two files. The record-matching process involves pairing records from the two files and determining whether a given record pair can be considered a match or a nonmatch. For any two files, there are always many more unmatched pairs than true matches. In two files, each of which contains exactly 500 records, the possible number of record pairs would be 500 × 500, or 250,000 possible record pairs. Because there are only 500 records in each file, the maximum number of matches one could hope to produce is 500 (assuming no duplicates in either file and perfect matches between the two files). The basic idea is to use common fields in both fields to match records. Each of the matching fields has certain properties that affect its performance as a matching variable. Some fields (e.g., date of birth, name, social security number) contain many different possible values. A match on one or more of these fields greatly increases the probability of a match between two records. On the other hand, many of these fields have a higher likelihood of errors and inconsistencies at the time data are collected and entered. Other fields, such as gender, zip code, political party affiliation, or other attributes with a limited number of possible values, may be more accurately entered but do little by themselves to increase the likelihood of matching record pairs. Of course, taken together matches on many individual fields help increase confidence of an overall match between records that have been paired. The matching algorithm involves determining the extent to which any individual field, as well as the summation of all fields used in matching, contributes to the probability of a true match.

FEATURES OF AUTOMATCH SOFTWARE

Automatch was recently developed and marketed by Matthew A. Jaro (MatchWare Technologies, Inc.) of Silver Spring, Maryland. Jaro is a computer scientist who left the U.S. Bureau of the Census to form a software development firm. Although MatchWare Technologies has many of the problems associated with small start-up ventures, one advantage of its small scale is that customers can deal directly with the developer. Slick packaging and carefully edited training manuals received from most vendors are less valuable than the personalized and informed user support received from MatchWare. There are not many users—in part because records linkage tends to be a more specialized field within social science research (few basic courses on data base management and statistical analysis include record linkage as a topic). Moreover, although there is tremendous potential for new uses and abuses of this technology, it is, for the most part, an emerging technology that has not yet been widely implemented.

Automatch is currently available in a PC version running on the MS-DOS or OS/2 operating system. It is also available in Unix versions for running on workstation environments. There are some differences between the PC and Unix versions of Automatch, but many are a function of operating system and hardware characteristics instead of program differences. Obvious differences are processing speeds and memory management. Although the PC version can run with just 640K, the performance is acceptable with only relatively small data bases. On the other hand, running Automatch on a workstation allows for the handling of much larger files. For example on a Sparc 10, a 70,000 record file was matched against a 9,000 record file in under 5 min.

In some respects, the PC version is more user-friendly than the Unix version tested. With a color menu–driven system, the PC version of Automatch can be used by most who are familiar with data base management systems. The PC version was found to be especially good for training purposes. Users must be able to define file structures clearly, name variables, specify types and lengths, and understand the basic principles of records linkage. If one could not carry out the records linkage manually, it would not be possible to instruct the machine to do so.

Automatch is a collection of specialized programs that operate by indexing instead of sorting the original files to be matched. To use Automatch, a certain amount of file preparation must be done. The amount of preparation will depend on the nature of the data collected as well as inputting, editing, error checking, and other data management practices. The files must be standard ASCII files, with each line delimited with a carriage return. Records must be of fixed size. Automatch does not support records or fields of variable length. Automatch will support most character, numeric, date, street address, and other types of variables. There is a procedure for defining missing value codes, although Automatch does not recognize the SAS use of "." as a missing value. Automatch is not a substitute for a data base management system or a statistical analysis system. Although one byproduct of a matching exercise is the identification of errors in the files being matched, Automatch is not equipped to correct or modify the original files directly. Automatch calculates certain statistics and distributions, which are specific to the matching procedures and not particularly useful for description, analysis, or modeling of data. The Unix version, unlike the PC version, is not at present a menu-driven system, so users must be familiar with a good text editor to write the control files required to run the program. Users of the Unix version of Automatch must have some elementary programming skills as small files are written and compiled to control the linkage process. Anyone who has written batch command files in DOS or has written control files in statistical packages such as SAS or statistical package for social sciences (SPSS) should have no difficulty mastering the Automatch system.

The program is designed so that users begin by assigning a project name that is used in all steps of the linkage procedure. The program generates various extensions that identify all the files for a given project. A first step in Automatch involves the creation of data dictionaries for the files to be matched. The data dictionary defines the location of the file, the record size, as well as variable names, positions, lengths, and missing value codes. The prepared dictionaries for the files are then compiled into binary format. It is impor-

tant to note that if changes are made to the dictionary or the underlying data set the dictionary must be recompiled.

In addition, users must prepare a match specification file to control the matching algorithm. It identifies the type of program, sets out a blocking scheme to subset the file, and lists matching variables and the matching procedure to be used with each one.

Automatch contains three different types of matching programs: (a) MATCH—for matching between records on two files, (b) GEOMATCH—for matching a file to a geographic reference file (e.g., the 1980 Census DIME map files or the 1992 Census TIGER map and street address files), and (c) UNDUP—for identifying duplicate records within a single file.

The Automatch procedure is efficient because it breaks the matching process into two distinct steps: (a) blocking the data in each file into small groups using a few variables that partition the total file into subsets of similar cases and (b) indexing the blocks and running the match comparisons within each block using probabilities of matches developed in the indexing stage.

In this way a 70,000 case file can be tested for possible matches against a 100,000 case file without examining 7 billion possible match pairs, that is, without actually testing every unlikely case. Auto crashes involving male 45-year-old drivers need not be tested against ambulance calls to pick up female 16-year-old injury victims. Thus, by restricting the range of comparisons to a block of plausible cases, the number of actual tests is dramatically reduced.

Blocking involves the creation of homogeneous subsets formed around variables such as age or place of residence. The more blocks that are created, the smaller they will be, and, therefore, the more efficient will be the matching procedure. Automatch recommends block sizes of 100 records per file. The PC version has a block limitation of 32,400 pairs in a block (180 records per block). The best variables for blocking are those with a large number of possible values and a high degree of reliability.

Automatch also requires the user to specify the variables to be used for matching and the cutoff values for declaring matches. The matching variables must be different from those selected for blocking. The program accepts a variety of different types of variables (character, numeric, time, odd or even interval, etc.). Depending on the type of variable selected for matching, different approaches to comparison are used. For example, with character fields, a character-by-character comparison is carried out with shorter fields padded with trailing blanks to match the length of the longer field. Automatch also provides for an uncertainty character field in which tolerance for phonetic errors, transpositions, random insertions, deletions, and other differences between two fields can be set. Numeric fields involve a straight algebraic numeric comparison in which leading spaces are converted to zeros and numbers are compared. This is particularly useful for record data that have ill-defined columns or out-of-place number values. Automatch also has a delta percent comparison in which differences between fields should be measured in percentages. There are also allowances for interval data and odd or even intervals (useful for geocoding applications).

After specifying the fields to be used for matching in terms of their names and types, the user must specify two different subjective probabilities, m and u. The m probability is the probability that the field agrees, given that the record pair is a match. The u probability is the probability that the field agrees at random. Although the user must provide an initial estimate of these probabilities, some guidelines are given that are helpful. It is easier to begin by estimating the u probabilities. For a field such as gender, where there are only two possible outcomes, male and female, the probability is .5. Estimating the probabilities for fields with more possible values, say, zip code or date of birth, may be more difficult, but it forces the user to think about the characteristics of the particular fields selected for matching. In a similar manner, the user should estimate the m probabilities. This probability can be estimated by subtracting the error of the field from one and typically ranges between .9 and .99. The prospects of having to estimate these probabilities may seem somewhat daunting, but Automatch contains a program that can be run to update this probability (after a match run is executed) that is based on the actual characteristics of data included in the matching files.

After specifying the probabilistic matching parameters, the user must also specify the cutoff weights that signify the threshold levels for an acceptable automated match and those cases that require clerical review. The weights for a given field are calculated by taking the log to the base 2 of the ratio of m and u probabilities (if the fields agree) and the log (base 2) of the ratio 1-m and 1-u (if the fields disagree). In this way, fields that agree receive positive weights and those that disagree receive negative scores. A composite weight for the record pair is calculated by summing all of the individual field weights. The program produces a histogram of these composite weights. Records that have a high positive weight are assumed to match and those that have a low or negative weight are assumed to be nonmatches. Based on the distribution for all comparisons, users are able to discriminate between matches and nonmatches (see Figure 1). On the basis of this distribution, cutoff weights can be established, and those cases that require clerical review can be identified.

Only occasionally will users be able in a single pass to determine the matching specifications and produce a satisfactory match. It is clear that, with each pass of the matching algorithm, more information about the data is gleaned and can be incorporated into the selection of blocking and matching variables as well as in the selection of appropriate cutoff weights. By design, Automatch is meant to be iterative; it may take several passes before initiating clerical review.

The clerical review process involves classifying record pairs as matches or residuals (nonmatching records). A report-generator program is built into Automatch to facilitate clerical review. This program enables the user to view records and additional fields defined in the data dictionary to make an assessment about whether a record pair is indeed a match. The clerical review program allows the user to examine not only potential matches but also duplicate records that may have ended up in either of the two comparison

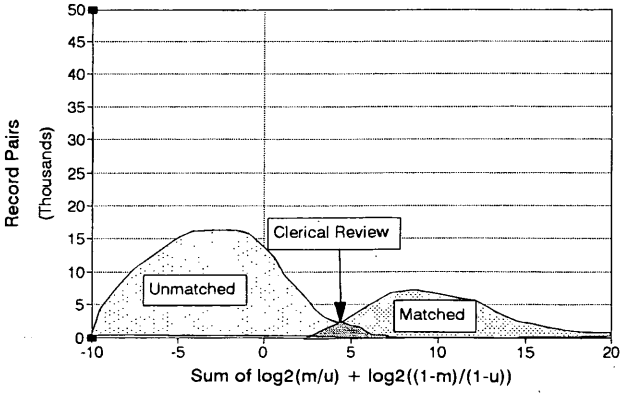

FIGURE 1 Profile of aggregate weights for matched and unmatched record pairs.

files. One feature of the clerical review algorithm is a history file, which keeps a record of all the decisions made by the clerical reviewer so that if the matcher program is rerun the user will not have to view the same records that have been previously reviewed.

The final step in Automatch generally involves producing an output file of the matched records that can then be imported into another package such as SAS or SPSSx for statistical analyses. Other special features are built into the Automatch program for handling geocoded data and producing other specialized reports related to the matching procedure. MatchWare Technologies has also developed address standardizers and other specialized programs and routines that are useful in file preparation and records linkage.

EVALUATION OF AUTOMATCH

In this section Automatch's performance on matching two data bases from Hawaii—the state Motor Vehicle Accident (MVA) file and the state Emergency Medical Services (EMS) Data Base—is described. Based on experience, an evaluation of Automatch's performance is provided. The data were matched as part of requirements for a federal grant to build a Crash Outcome Data Evaluation System (CODES), administered by the Department of Transportation. The Hawaii CODES project involves linking crash data to ambulance transport (EMS) data, hospital data, insurance data, and other information on traffic crashes in Hawaii during 1990. The purpose of the project is to build a linked data base on which models explaining crashes, driver behavior, and the effectiveness of safety devices on reducing injury and fatality can be tested.

The MVA data contains computerized records on all major traffic accidents in Hawaii. Data are collected by police officers called to the scene of a traffic collision. Data on driver, vehicle, occupant, and roadway characteristics are filled out on a paper form, which data in turn are entered into a computer system maintained by the Department of Transportation, Hawaii. In 1990, there were approximately 27,000 major traffic crashes, involving some 45,000 drivers, and an additional 30,000 occupants. The data suffer because it is collected by police officers under not ideal conditions and then entered by keypunchers who have few resources with which to check or verify the work. On the other hand, there are only four counties in Hawaii, and the data are more centralized than in many other states with more local law enforcement agencies.

The EMS data are also collected on a statewide basis and maintained by the Department of Health, Hawaii. This data base draws information from a dispatch card, which is filled out when a call for ambulance service comes in, and an EMS report, which is completed by ambulance attendants called to the scene of an accident. When the nonemergency, nontraffic-related ambulance runs are excluded from the EMS data base, approximately 9,000 ambulance runs must be accounted for.

The steps in records linkage involve cleaning the two data bases, preparing the fields for matching, devising a matching strategy, running several passes with the Automatch program, conducting clerical review, and preparing various summary reports.

In matching EMS records to MVA records, more time and effort went into the preparation, editing, and cleaning of the files than in conducting the matching procedure. Part of the reason for this has to do with the nature of public data bases that are maintained more for individual records reporting than for data analysis and modeling. A basic hardware problem involved extricating and decoding the MVA data from an antiquated Wang minicomputer system before it could be read on to the Sparc 10 workstation. For the MVA data, the data files were prepared, using SAS, by recording variables into a usable format and writing them to a text file using the "PUT" command. For the EMS data, dBASE was initially used because the data had been entered into a relational data base management system. Eventually, the data were transferred into SAS so that comparable matching variables could be constructed.

The blocking strategy was dictated by the nature of error in the original data. Blocks were to be as small as possible to provide near certainty that matches that were not physically possible would be prohibited. The most important variables for blocking were county and date. These were good variables because in Hawaii each of the four counties consists of separate islands, isolated by ocean. The date field was systematically edited and verified by EMS personnel. We also used gender as a blocking variable. These blocks enabled a match with greater certainty on the variables such as age, time of the incident and service, and location codes. The matching strategy produced a distribution of matched and unmatched pairs, including exact matches, duplicates, and clerical (manual) review cases (see Figure 2).

At the outset, it is important to note that Automatch's performance was impressive. First, few products comparable in cost or flexibility are available. Second, there is an underlying mathematical basis for the matching algorithm that is based on probability theory and enables the user to specify error ranges and accompanying levels of tolerance. Also the user is led through a logical sequence of data definition, developing a blocking and matching strategy, adjusting or correcting the strategy based on information generated through the match procedure, and can set parameters for clerical review. Automatch encourages the user to think systematically about the data that are being matched. Third, the level of technical support and the quality of customer service offered by MatchWare Technologies, Inc., has been superior.

```
**************************************************************
*
* OUTPUT STATISTICS FOR MATCH: sec
* PASS: 1
*
*    69072      Records on file A
*     9395      Records on file B
*        0      A residuals from previous pass
*        0      B residuals from previous pass
*    65795      A records read
*     9334      B records read
*     2012      Blocks processed
*        0      OVERFLOW blocks
*      173      Maximum A block size (including overflow)
*     30.8      Average A block size (not including overflow)
*       29      Maximum B block size (including overflow)
*      4.6      Average B block size (not including overflow)
*     6786      Matched pairs
*      227      EXACT matched pairs
*      537      Clerical pairs
*     4144      A duplicates
*        4      EXACT A duplicates
*      171      B duplicates
*        2      EXACT B duplicates
*    57605      A residuals (including skips & missing)
*     1901      B residuals (including skips & missing)
*     3745      A records skipped
*       30      B records skipped
*
**************************************************************
```

FIGURE 2 Sample output statistics from Automatch.

The documentation is clearly written and provides enough for most users to start using the Automatch software, but the documentation is thin (approximately 60 pages) and may not be enough for those who are doing records linkage for the first time. The program has been used several times, and the documentation now appears all the more clear and straightforward. The documentation falls short because no example is worked out all the way from start to finish with all inputs statements and screen outputs. The documentation tends to be one-sided, as it provides fairly good instructions in terms of statements and commands but leaves out the system responses, which, unless one has been through the entire matching procedure, are not the most informative. In using the system, the most common response was, "Now what?" In the spirit of DOS and Unix, in Automatch no response is a good response.

Another area with some degree of mystery involves the generation of program files and object files. Automatch generates several different files, so it would be useful to provide a more clear discussion of what files are created and how each is used in the system. The directory contents and watches were checked periodically to determine the effects of various programs and match runs to determine what new files were being created and modified. One would also like to have more information about the actual matching algorithm. Although Jaro's work (*5*) provides a basic understanding of what is going within the program, the documentation does not rehearse the algorithm in the context of a worked-out problem. More discussion is needed about the various user decisions that influence the matching procedure. Here too, an example or two worked all the way through from beginning to end, replete with the determination of composite weights and cutoff scores, would help bridge the gap between documentation and implementation. The concerns about the documentation are minor because these appear easily correctable deficiencies. Jaro (*5*) provided excellent technical support when needed.

Areas in which there is more room for improvement are user interfaces, screen calls, and the transitions between one program and the next. Although the PC version, with its menu-driven format is more user-friendly than the Unix version used, user interface support can be improved. It would be nice, for example, to have pull-down menus with program templates to serve as guides not only for writing individual programs but also for showing the sequence from one step to the next. Error messages could be improved so that debugging would be easier. Screen prompts emerge when submitting and executing commands, but many program files are prepared in batch format. When a program bombs, it is sometimes difficult for new users to figure out from error messages what went wrong. It would be useful to build a program editor into Automatch specific to the Automatch language so that illegal entries and inconsistencies would be flagged before compilation of the program.

Once one has a basic understanding of the principles of records matching and how to interpret the information provided by Automatch, then the actual records linkage becomes more challenging. One learns how to use Automatch by formulating blocking strategies, identifying matching variables, estimating the m and u probabilities, adjusting cutoff weights, and working the data bases to minimize the number of clerical reviews and retaining high confidence that the algorithm has done a good job of matching.

Special care is required in any form of raw-data procedure in which the underlying data are thought of as variables or measurements and would normally be indexed by named variables in a statistical analysis system or data base manager. This applies here as well. If the data sets to be merged are small and contain relatively few distinct variables, it is possible to create files by merging that encompass entire records. If the records are large, however—100 or 200 field records—it will be more efficient to extract a subset of variables that constitute the blocking, matching, and ID indicators for the set required to marry the subset back to the original records exactly. The user is tempted simply to use the sequence number of the record as the key for this purpose. This is not advisable for at least two reasons: (a) The order of the key is dependent on the order of the original file; if the file is transformed or, worse, sorted, the order is lost. (b) Any attempt to match the "unmatched" cases on a second pass, once the original matched records have been merged into a new, combined data set, is likely to define the "unmatched" cases as a subset of the original cases. The positional ID number of cases in the subset will be different from the ID number in the original file. (It is possible to execute successful matches on, say, 8,000 of 64,000 records from File A to File B, then return to try to match the 56,000 unmatched records from File A. To bring the results of this second matching process back into the master data set and to marry the cases correctly, unique identifiers must be carried into the subset used in the matching procedure.

It is easy to avoid the traps of this procedure by executing multiple blocking and matching steps within a single run of Automatch. In this way the criteria for matching can be upgraded on the basis of experience or new information, yet only one process will be used to merge the matched files back onto the master data base file.

Though Automatch has some rough edges, the product is fairly easy to use. Most computer-literate individuals can master the program in a few hours, provided that they start with a simple matching problem (with many good fields for comparison) and then graduate to a more difficult and realistic matching exercise. The flexibility of the program allows matching of different types of data sets and includes many special features that emerge more as one interacts with it. It is likely that most data base managers would find Automatch to be something that, once used, would be difficult to live without.

Other Applications and Uses

Automatch was developed for postenumeration surveys conducted by the U.S. Bureau of the Census. Undercounting of certain groups (minorities, non-English speakers, etc.) is suspected in some areas. Follow-up surveys are typically conducted in these areas. Automatch enables the comparison of individual records (between the original and follow-up survey) to find out which people were not counted the first time but were enumerated the second time around, to produce an estimate of undercounting.

Follow-up surveys, longitudinal questionnaires, and other applications that involve matching pairs for study over time could benefit from the use of a program such as Automatch. This is particularly useful when errors in data entry or substantial changes in population characteristics over time are concerns (*6*). Automatch enables matching to go beyond merely the use of one or two identifiers and permits many different kinds of variables to be used in records matching.

Another procedure in Automatch that identifies duplicate records would have many potential applications, from purging mailing lists of duplicates to removing duplicate records before updating a data base or performing statistical analyses on it.

The geocoding applications involve matching a particular data file with a reference file. For example, one could match data on

crash locations (typically coded in terms of street name and mile marker or cross street) with latitude and longitude data from computerized street index files or geocoded reference files. As Geographic Information System (GIS) technologies continue to expand, more reference files (e.g., Census 1992 TIGER map files, Census Summary Tape File data organized by ZIP codes, etc.) have become available in a variety of formats. Automatch can be used with GIS and mapping technologies to bring this sort of summary data into a format suitable for mapping.

For data base management, Automatch may be particularly useful in those circumstances in which there is much transaction processing. Organizations with large data bases in which information is continually being updated and altered may find use for probabilistic matching for error detection and postaudit review. At present, Automatch is set up only for batch processing, yet one could imagine ways of applying the algorithms in a more interactive fashion.

Other applications for Automatch may be in the field of criminal justice research, where one could examine the relationship between, say, traffic citations, traffic collisions, criminal activity, and other forms of deviant behavior (e.g., DUI, drug use, etc.). In the future, firms specializing in records linkage might emerge—similar to the emergence of those that provide geocoding services in response to the growth of geographic information systems. Many different applications can be envisioned in urban and regional planning, for example, linking transportation data to land use and marketing studies, social services data to census data, and environmental quality studies to data on land use and ownership.

CONCLUSIONS

Automatch does much to elevate the level of sophistication of data base managers and others who enter, clean, and match data. Too often, the business of data base management and records linkage has been kept in the dark ages—that is, although there is much statistical, graphics, and presentation software, really new tools in data base management have been rare. Automatch is an exception. It provides data base managers a new arsenal of programs for matching data, identifying duplicate records, and handling assorted problems typically associated with geocoded data.

Automatch also opens doors for researchers and statistical modelers looking for ways of combining data bases. Through records linkage, new and interesting analyses can be carried out. Gaps among agencies, disciplines, time periods, and data sources can be bridged through records linkage. The potential uses and abuses of this technology are great. The prospect of linking specific public record data bases (property ownership or voter registration files) to attitude survey, market research, health, or financial reporting data bases presents enormous ethical and political challenges. With this software and some understanding of probabilistic records linkage, even files in which many of the common identifiers (e.g., name, social security number, etc.) have been stripped can be linked to other public data files (which could contain names, addresses, and other person-level identifiers). Although this paper is meant to provide an overview of the technology, there are also important questions of what constitutes appropriate and legal data linkages and major questions about maintenance of confidentiality and the uses of data for purposes other than for what they were collected. Certainly, one response to the technology may be to make it more difficult than ever to gain access to computerized data.

A more complete discussion of the ethics of records linkage must come before widespread application of this technology—although as often happens with innovation, progress often precedes policymaking. Records linkage is definitely on its way to becoming a more widespread practice. For planners to appreciate its potential and limitations more fully, more systematic discussion about appropriate plans, policies, and standards of practice for automatic records linkage must occur. Educators have an especially important role to play not only in teaching the technology of records linkage but also in conveying a critical understanding of ethical concerns as well.

Because of the existence of probabilistic matching software, real data hounds will undoubtedly discover ways of improving the quality and coverage of information that will only serve to improve and expand upon the nature and levels of analysis and model building. MatchWare Technologies, Inc., has already developed a small impressive set of clients, ranging from the U.S. Department of Transportation to various public- and private-sector organizations around the world. We predict that Automatch—in its present form and versions beyond—will become more widely used and that the practice of probabilistic records linkage, with all its opportunities and challenges, is here to stay.

ACKNOWLEDGMENTS

Automatch is available from MatchWare Technologies, Inc., 14637 Locustwood Lane, Silver Spring, Maryland 20905. The program was tested and run on a Sun Sparc 10 workstation at the Department of Urban and Regional Planning, University of Hawaii. Support for this research was provided by the Hawaii CODES Project, a cooperative research agreement between the U.S. Department of Transportation, National Highway Safety Administration, and the University of Hawaii. The authors acknowledge the support of the Department of Transportation, Hawaii, and the EMS Branch of the Department of Health, Hawaii. Graduate students in the Department of Urban and Regional Planning at the University of Hawaii (Richard Kirschenbaum, George Nabeshima, and John Valera) helped in the preparation of data files used in this procedure.

REFERENCES

1. *Automatch: Generalized Record Linkage System, User's Manual, Version 1.4*. MatchWare Technologies, Inc., Silver Spring, Md., 1993.
2. Felligi, I. P., and A. B. Sunter. A Theory for Record Linkage, *Journal of the American Statistical Association*, Vol. 64, 1969, pp. 1183–1210.
3. Newcombe, H., and J. M. Kennedy. Record Linkage, *Communications of the Association for Computing Machinery*, Vol. 5, 1962, pp. 563–566.
4. Newcombe, H. *Handbook of Record Linkage: Methods for Health and Statistical Studies, Administration, and Business*. Oxford University Press, Oxford, England, 1988.
5. Jaro, M. A. Advances in Records-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, Vol. 84, No. 406, 1989, pp. 414–420.
6. Newcombe, H. B., M. E. Fair, and P. Lalonde. The Use of Names for Linking Personal Records, *Journal of the American Statistical Association*. Vol. 87. No. 420, Dec. 1982, pp. 1193–1024.

Publication of this paper sponsored by Committee on Traffic Records and Accident Analysis.