Comparing BigMatch Results to Current National Death Index (NDI) Selection Methods

Bryan Sayer [1]
Social & Scientific Systems, Inc.

Abstract

Using a frequency of sex specific first names and middle initials, plus non sex specific last names from 22 years of the NDI (about 48 million people) we generate an artificial master database that replicates the general NDI structure, and an artificial matching file that contains both decedents and non-decedents. The matching file also contains ‘altered’ submission records that represent the types of problems found in survey records that are matched to the NDI, such as miss-spelled first names and middle name substituted for first name. We compare the results of using a variety of relaxed blocking assignments in Big Match to estimated results under the current system. We show which correct records are selected by one, the other, or both of the systems, and which system works the best with which types of altered records. In addition, we show that BigMatch can successfully use limited information (such as just the last four digits of social security number (SSN) or the four characters of first name) that the present system cannot replicate.

Keywords: Record Linkage, National Death Index, Artificial Data, Probabilistic Matching

1. Basics of NDI Matching

1.1 Current System

The current record linkage system for the National Death Index (NDI) runs in ADABASE on the CDC Mainframe in Atlanta, GA (Bilgrad). It works by pre-indexing both files (the NDI master file and the user matching file) and comparing on a set of seven specific blocking factors. All NDI records that meet any one of the seven criteria are selected as potential matches and output for the user to evaluate. The blocking factors use a modified version of the New York State Identification and Intelligence System (NYSIIS) sound alike system for last name matching, but first name must match exactly. Each potential match is then scored using a probabilistic scoring system developed by NCHS staff in the early 1990s(Horm A-5-A-13). The seven blocking factors in current use are:

1. Social Security Number (exact match on all nine digits)
2. Exact month and +/- 1 year of birth, first and last name
3. Last name, first and middle initials, month of birth, year of birth (plus or minus 1 year)
4. First and last name, exact month and day of birth
5. Last name, first and middle initials, exact month and day of birth
6. Exact month and year of birth, first name and birth surname
7. Exact month and year of birth, first name and last name on user’s record to birth surname on NDI record

Although all potential matches are output for the user, the actual identifying information on the NDI record is not included in the output. Only an indication of whether the item on the NDI record (such as first name) matches the information on the user’s record. Hence if the user’s record contains a first name of Beth, and the NDI record contains a first name of Elizabeth, the user does not get to see the NDI first name value of Elizabeth. The user only gets an indicator showing that first name does not agree on the two records. This is done for confidentiality reasons.

Since the current NDI system works on a specific set of pre-defined blocking factors, items on the user records must be complete (or complete except for Social Security Number – SSN) in order for the system to work. Variations such as having only the last four digits of SSN or the first four characters of first name will not work in the current system(Sayer and Cox).

1.2 BigMatch

BigMatch is a record linkage program from the U.S. Census Bureau that compares two files of records where some of the records in each file may be for the same entity(Yancey). BigMatch works similarly to most other record linkage programs(Eugene Rogot., Paul Sorlie, and Johnson 719-34;Fellegi and Sunter 1183-210) in that blocking factors are described which must match in order for a record pair to be considered a potential match. Unlike most other record linkage programs, BigMatch does not require the two files to be indexed in advance and, perhaps most importantly, BigMatch puts the indexes it builds into memory making the comparisons extremely fast and allowing the pool of potential matches to be very large.

[1] The author wishes to acknowledge the contributions of Jay J. Kim (CoCHIS/CDC/NCHS/ORM) and Mike Larsen (Iowa State University) for helpful comments regarding this work. This work is the result of contract 200-2005-M-13283 with CDC/NCHS. Any opinion stated herein is the author’s and should not be attributed

BigMatch does not attempt to make a decision regarding the match status(Yancey). Rather it compares record pairs that match on the blocking criteria, and then keeps as a possible match all pairs that meet the cutoff standard. More specifically, BigMatch discards any record pair whose total score does not meet a value specified as an option for the program.

Therefore, there is not a way to compare "true matches", "false matches", etc. between the current system and BigMatch. Instead, the comparison is the number of correct records found by each method, stratified by the "type" of submission record. However, there may be one other interesting feature of BigMatch which involves the probability cutoff standard. One issue with the current selection system is that it returns to users all the records that meet any of the seven blocking criteria. Following the reasoning of "at most one match per person" as a way to reduce the number of records returned to users, there may be a way to rank the potential matches using the score generated by BigMatch. If there exists a ranking system in which the correct match always "rises to the top", this would yield a substantial positive improvement over the current system. Note that in the case of ranking, many non-decedents would still have a record that was top ranked, but would not represent a correct match. This is the fundamental difference between a scoring system that would attempt to identify true matches and a ranking system.

2. Data Generation

2.1 Source Data and Program Operation

For most of the items on the records we use frequencies of each of the twelve characteristics included on the NDI record from years of death 1979 through 2000 inclusive. A few of the items are sex specific (first name, middle initial) and marital status is both sex and age (10 year age groups) specific. Most of the twelve items include all possible values, with the probability of day of birth fixed at 1/30 for days 1 to 31. First and last name include a subset using the most common values. For last name (and birth surname which is a copy of last name), this is the 10,000 most common. First name uses only the 250 most common sex specific values. Middle name uses the probability distribution of middle initial from the NDI data, but picks a middle name with that initial from the first name data. Year of birth is restricted to 1901 to 1920. These last two limitations are designed to make the master records more "common", that is increase the number of possible matches in order to not have a high degree of discrimination from very rare first names or be able to distinguish false matches by year of birth alone.

The twelve items are:

1. Social Security Number (9 positions, with each position having a separate weight)
2. First Name (Year of Birth and Sex specific)
3. Middle Initial (Sex specific)
4. Last Name (and Birth Surname) (10,000 most common)
5. Race
6. Sex
7. State of Birth
8. State of Residence
9. Marital Status (sex and age at death specific)
10. Month of Birth
11. Year of Birth
12. Day of Birth

Each weight includes the probability of occurrence based on the available items. For items such as last name that include only a subset of the available values, the total reflects the sum of the occurrences of the subset rather than the entire file. In general the total is between 46 and 47 million. Last name has 32.3 million (meaning the 10,000 most common last names account for 32.3 out of 47 values). The first name values account for 42.4 million records, though the ten most common male names account for nearly 28% of the male total, while the ten most common female names account for only about 18% of the total.

For the artificial data, we created a SAS program that used the scoring files as input. The program loops once for each sex and once for year of birth. In this specific case we looped 40 times (2 sexes times 20 years of birth). In general, it uses this approach:

1. The total number of persons (both dead and alive) is chosen (1,000,000)
2. The number of people in the matching file is chosen. (10,000)
3. The proportion of the matching file that is to be deceased is chosen (50%)
4. Year of birth is fixed to the years 1901 to 1920.
5. Age at death is set to approximate the age at death for deaths occurring in 2000. This fixes the year of death, based on age at death and year of birth
6. Month and day of death are randomly generated, with month of death restricted to 1 to 12, and day of death restricted to valid values for the month.

7. Sex specific first name (restricted to the 250 most common for each sex) is selected.
8. Middle initial is selected, sex specific. For a full middle name, we use the first name file as a source, based on the initial from the middle initial file and then getting a name from the first name file having that initial. We add middle name onto the end of the record, as it is not currently used.
9. Last name is selected.
10. For non-single females, birth surname is selected from the last name fil.e
11. Race from the race probabilities.
12. State of birth from the state of birth probabilities.
13. State of residence from the state of residence probabilities.
14. Social Security Number – each digit is generated from the relative frequencies.

To do these tests we have created our own versions of the NDI master file and a typically user's file, with variations. The master file contains approximately 1,000,000 records and is loosely based on the probability distributions of each item from the NDI data for 1979 through 2000.

2.2 Master File Data (Death Index)

From this data, an ASCII record is produced with the same layout as the NDI file, with the addition of middle name and a special ID number. Sound alike (NYSIIS) values of first, middle, last, and birth surnames are also added.

2.2 Matching Data (User's File)

The baseline record is the exact match record from the master file for decedents, and similarly for non-decedents, except that there is no corresponding master record. That is, a non-decedent's record is created exactly the same as all other records, but is not written to the master file. Date of death is interpreted as date of interview for non-decedents. In addition to the baseline record, and based on input from a variety of sources, we develop a number of "error" records for use as alternate submission records. Due to a variety of reasons including fear of identity theft and changes in the questionnaire wording, we know that that survey respondents are increasingly reluctant to divulge SSN (for example, the NHIS now has a 60% refusal rate), so we test using only the last four digits of SSN and no SSN. From past research we also know the common reasons for missing a known death. Most common are problems with first name, as currently it must match exactly. If it does not, first initial and middle initial must match. With the large number of missing middle initials in the NDI file (about 25%) mistakes on middle initial are common and cause deaths to be missed when first name is even slightly miss-spelled. The changes to first name are created by substituting a completely different first name, which does exaggerate first name problems. In particular, miss-spellings of first name often have a correct first initial, and this combined with a correct middle initial is sufficient to select a correct record. For all eligible records, a corresponding "error" record is generated. However, not all original records are eligible for all types of error records. In particular, about 25% of master file records have a blank middle initial (as does the real NDI). Hence these records do not have a middle name and are not eligible for middle name variations. The following list indicates the types of error records we test. Counts of each type of record for decedents are in tables 1 (males) and 2 (females).

1. Original Record.
2. Last Four Digits of SSN.
3. No SSN.
4. Change First Name.
5. Last 4 Digits of SSN, Change First Name.
6. No SSN, Change First Name.
7. No Middle Initial, Change First Name, Change MOB.
8. No SSN, Change First Name, Change MOB.
9. Females Only-New Last Name, New Marital Status.
10. Females Only-New Last and First Name, New Marital Status, state of residence, 4 digit SSN.
11. Females Only-First Four Characters of First Name, Four digit SSN.
12. Males Only-New Marital Status and new First Name, Four digit SSN.

Results

All tests are run on a Dell D600 laptop and generally ran in 15 to 75 seconds. Each record type is run as a separate run, though males and females of each record type are combined. The maximum number of user matching records is 10,000. The master file remains constant and has nearly 1 million records.

Results of selected runs for both the current system and BigMatch are shown in tables 1 (males) and 2 (females) by different types of "error" records (numbered for convenience). The first column of numbers shows the count of true records that should be found. The next two columns show what happens under the existing system (one including SSN matching, one excluding SSN matching). The last four columns show the number of correct records selected by BigMatch, where the correct record ranks using the score, the blocking factors used, and the fields used for scoring. The majority of these runs use a bottom (rejection) criterion of zero, meaning that all records retrieved are kept. However, for test purposes, one record type (9) uses a bottom threshold of 4.7 and includes an "incorrect" factor in the scoring equation. This results in 109 correct records being dropped for females, which is a result of the "incorrect" first name.

Since the types of errors are known in advance, it is easy to define blocking criteria in BigMatch that sucessfully retrieve the correct record. Ranking all of the potential matches so that the correct record ranks first is more difficult. In particular, if a non-matching field is included in the total score then many times an incorrect record will rank higher than the correct record. Examples of this include record types 3 and 5. For example, for males with record type 3 (incorrect first name) only 1,711 of 2,567 correct records rank first, when including first name in the set of fields used for scoring. Perhaps more importantly over 450 correct records have a rank of 4 or more indicating that it is not possible to count on the correct record being in even the top 3 records under these specifications. The impact is even greater for femles.

BigMatch does retrieve records that are missed under the present system, and can operate easily on less than complete information. The issues of not dropping the correct record through too high of a bottom threshold and getting the correct record to "rise to the top" are more difficult. As these records are specifically designed to represent the types of errors that we know causes the existing system to miss finding the correct record, it is not surprising that often many correct records are missed, once SSN is excluded. Since the structure of the submission record is known, it is always possible to find (and keep) all correct records if the proper blocking factor is invoked and the cutoff is set low enough. However, sometimes this results in a very large number of incorrect records are also retrieved.

One of the specific scoring issues involves whether incorrect fields are included in the scoring. If only fields that are in agreement are used for scoring, then the correct record is always the top scoring one. However when fields not in agreement (types 3, 5, and 11, for example) are included in the scoring, the correct record can fall in rank a great deal in the scoring. In addition, when the amount of information in a field is shortened (four characters of first name – types 17) or the distinguishing power of the fields is low (type 14) the correct record is often not the highest scoring record (or is not the only record with the highest score).

Some additional work not shown in the table includes changing the ratio of the match probabilities. The results indicate that altering these probabilities is not particularly helpful in eliminating non-matches while keeping correct matches. This is not surprising given the crude nature of these probabilities(Yancey).

Scoring the fields used for blocking factors does not add any discriminating information. Since any record pair must match on the blocking fields, using the same fields for scoring is redundant. All records earn the same points, so scoring them has no distinguishing value.

BigMatch can easily handle fields with partial information, but as it is currently programmed, all records have to be the same. That is, it is not equipped at present to handle a file where some respondents have given their entire SSN, some have given only their last four digits, and some have refused completely. Such a file can be separated into records of each type and runs made on the individual files, but we believe that it is possible to actually modify the BigMatch program to change this and allow BigMatch to accommodate mixed files.

Conclusion

That BigMatch can select correct records missed by the current system, and work with shortened information, is clear. These tests show that based on blocking factors alone, BigMatch is more flexible and able to find correct matches using less information than the current system. It would be possible to modify the program to scan incoming user records for missing or incomplete fields and have BigMatch use only the then relevant blocking and scoring factors, which would eliminate the need to separate the different types of user records and have multiple runs. Although not used much in these tests, BigMatch allows multiple blocking factors in a single run, and this could substitute for the separate runs.

The principal remaining question is still "is there a method to sort the retrieved records where the correct record (if present) always (or nearly always) rises to the top." Specifically, with regards to scoring, would a scoring system that attributed positive points to fields in agreement, but does not subtract for fields in disagreement make the correct record "rise to the top"? Would scoring all (or most) of the fields mitigate the impact of the incorrect fields?

An alternative approach is to try and eliminate non-correct matches, rather than attempt to force the correct match to rise to the top. This may be possible by scoring more of the fields than attempted in this test, perhaps eliminating any penalty for non-agreement, and adjusting the bottom cut-off to eliminate non-matches.

The final issue, not addressed here, is identifying the correct record. It may be that a Bayesian scoring system could be added to BigMatch that would distinguish true matches from false matches under these varying circumstances. The big advantage to a scoring system being added to BigMatch is that it will reduce the number of records returned to users to at most one or two per subject. The current system outputs all retrieved records before scoring them, hence the number records returned to users is very large, relative to the number of true deaths (about 10 incorrect records for each correct record)(Sayer and Cox). Reducing this to one or two will greatly simplify the user's job of sorting through the records to find the correct matches.

References

Eugene Rogot., Paul Sorlie, and Norman J. Johnson. "Probabilistic Methods In Matching Census Samples To The National Death Index." Journal of Chronic Disease 39.9 (1986): 719-34.

Fellegi, Ivan P. and Alan B. Sunter. "A Theory for Record Linkage." Journal of the American Statistical Association 64.328 (1969): 1183-210.

Horm, John W. Assignment Of Probabilistic Scores To National Death Index Record Matches. A-5-A-13. 12-1-1996. Centers for Disease Control and Prevention/National Center for Health Statistics. National Death Index Plus: Coded Causes of Death.
Ref Type: Report

Sayer, Bryan D. and Cox, Christine. S. Zombies, Immortals and the Which Hunt. 6-14-2002.
Ref Type: Unpublished Work

---. "How Many Digits in a Handshake: National Death Index Matching With Less Than Nine Digits of the SSN"., August 3, 2003: 2003 ASA Proceedings. Alexandria, VA: American Statistical Association, 2004.

Yancey, William E. BigMatch: A Program for Extracting Probable Matches From A Large File. 5-6-2004.
Ref Type: Computer Program

Table 1: Males, Results of Existing NDI Retrieval System and BigMatch

	True Decedents With Record Type	Current System – Number of Correct Records		Big Match - Number of Correct Records		Big Match Options Specified	
Type of Record		All 7 Selection Methods	Other Than SSN	**Rank**	**Count**	**Blocking Factor(s)**	**Scoring Factors**
0:Exact Record	2,567	2,567	2,567	1	2,567		
1:4 Digits of SSN	2,567	n/a	2,567	1	2,567	Last 4 Digits of SSN	
2:No SSN	2,567	2,567	2,567	1	2,567	Last name	
3:Change 1st Name	2,567	2,567	123	1	1,711	NYSIIS Last name, Sex	Last name, First name, YOB, SOB
				2	328		
				3	169	NYSIIS Birth Surname, Sex	Last name, First name, YOB
				4+	359		
4:4 Digits of SSN, Change 1st Name	2,567	n/a	123	1	2,567	Last 4 Digits of SSN, Sex, NYSIIS Last Name	Last name, First name, YOB, SOB
5:No SSN, Change 1st Name	2,567	123	123	1	1,711	NYSIIS Last name, Sex	Last name, First name, YOB, SOB
				2	328		
				3	169		
				4+	359		
8:No M.I., Change 1st Name, Change MOB	1,944	1,944	0	1	1,944	NYSIIS Last name, Sex	Last name, First name, Middle Initial, YOB, MOB
9:No SSN, Change 1st Name, Change MOB	2,567	n/a	0	1	1,944	NYSIIS Last name	Last name, First name, Middle Initial, YOB, MOB
23:M-New MS and 1st Name, 4 digit SSN	1,085	n/a	0	1	1,083	NYSIIS Last name, Last 4 digits of SSN	First name,YOB, Last 4 digits of SSN
				2	2		

Table 2: Females, Results of Existing NDI Retrieval System and BigMatch

	True Decedents With Record Type	Current System – Number of Correct Records		Big Match - Number of Correct Records		Big Match Options	
		All 7 Selection Methods	Other Than SSN				
Type of Record				**Rank**	**Count**	**Blocking Factor(s)**	**Scoring Factors**
0:Exact Record	2,476	2,476	2,476	1	2,476		
1:4 Digits of SSN	2,476	n/a	2,476	1	2,476	Last 4 Digits of SSN	
2:No SSN	2,476	2,476	2,476	1	2,476	Last name	
3:Change 1st Name	2,476	2,476	149	1	1,236	NYSIIS Last name, Sex	Last name, First name, YOB, SOB
				2	343		
				3	195	NYSIIS Birth Surname, Sex	Last name, First name, YOB
				4+	702		
4:4 Digits of SSN, Change 1st Name	2,476	n/a	149	1	2,476	Last 4 Digits of SSN, Sex, NYSIIS Last Name	Last name, First name, YOB, SOB
5:No SSN, Change 1st Name	2,476	149	149	1	1,236	NYSIIS Last name, Sex	Last name, First name, YOB, SOB
				2	343		
				3	195		
				4+	702		
8:No M.I., Change 1st Name, Change MOB	1,835	1,835	0	1	1,835	NYSIIS Last name, Sex	Last name, First name, Middle Initial, YOB, MOB
						NYSIIS Birth surname, Sex	Last name, First name, YOB
9:No SSN, Change 1st Name, Change MOB	2,476	n/a	0	1	1,835	NYSIIS Last name	Last name, First name, Middle Initial, YOB, MOB
11:New Last Name, Marital Status	1,813	0	0	1	405	NYSIIS Last name	First name, YOB, Marital Status
				2	151	NYSIIS Birth surname, Sex	NYSIIS Birth surname, First name, YOB
				3	120		
				4+	1,134		
14:New Last and 1st Name, MS, SOR 4 digit SSN	1,813	n/a		1	1,477	Last 4 Digits of SSN	Sex, YOB, State of birth
				2+	336		

17:4 Char.1st Name, 4 digit SSN	1,813	n/a	na/	1	572	Last 4 digits of SSN, First 4 characters of first name	First 4 characters of first name, YOB, State of Birth
				2	209	NYSIIS Birth Surname, Sex	NYSIIS Birth surname, YOB, First 4 characters of first name
				3	136		
				4+	896		