Link the death and CHARS files

Eric Ossiander
July 17, 2014

I created linked death-hospitalization files for deaths occurring in the years 2010, 2011, and 2012. I linked the deaths occurring in each year to the hospitalizations in that year and the previous year.

I used the following fields in the linking process:

birth date
name
last 4 digits of SSN
sex
zipcode of residence
county of residence
hospital code
death date
Hispanic ethnicity
race
state of residence

I used the RecordLinkage package in R for most of the linking. In all of the record linking that I did in R, I used birth date as a blocking field (i.e. I required that the birth date on the death certificate match the birth date on the hospitalization record). First, I computed a probabilistic linkage weight for each record pair. Second, I used a machine learning algorithm to predict which record pairs were links. (This required me to manually code a training set once, to create a statistical model for predicting links. Then I used the statistical model for each subsequent year of data.) Then I manually reviewed all of the record pairs which were predicted not to be a link by the machine learning algorithm, but which had a high probabilistic weight, and all record pairs which were predicted to be a link, but had a low weight. I also used a SAS program to compute a probabilistic linkage weight for all record pairs (i.e. not blocking on birth date), and manually reviewed all of the record pairs that had a high probabilistic weight in which the death certificate birth date did not match the hospitalization birth date. I combined the three linked sets (the machine-linked pairs, the manual review of the machine linking, and the manual coding of the non-birth date matching pairs). Then I checked for hospitalization records that linked to more than one death record, and manually adjudicated those links.

■ ■ *begin analysis details*

Process

Death file

Items that help identify people in the death file

name
dob
social security number
age at death
date of death
sex
race, ethnicity
place of residence (address, county, zipcode)
place of death (county, city, facility, facility type

These items are split between the public use file and the death names file, so I will need to combine those files and extract the relevant fields.

CHARS file

items that help identify people in the CHARS file

name
dob
social security number (last 4 digits)
age
sex
race, ethnicity
discharge date
discharge status
hospital code
place of residence (zipcode and county)

All of these items are in the confidential files (names `chr_r2012.sas7bdat`, etc).

It looks like names are present on a few records in 2008, and on almost all records in 2009 and following years. In 2007 and earlier (and in the 2008 records that don't have names), first two letters of names are on the files. Birthdates are apparently on all files. SSN is two-thirds missing in 2008, better in 2009, and about 20% missing in 2012. It is almost entirely missing in 2007. Race is reported on about 40% of 2008 records, very few before that year, and almost all records after that year.

linking

If the availability of birth date is good on both files, I will study whether that can be used for blocking. After that I will probably use these for linking:

death	CHARS
name	name
first 2 chars	first 2 chars
soundex(name)	soundex(name)
SSN (last 4)	SSN
sex	sex
race	race
hispanic	hispanic
zipcode of res	zipcode of res
county of res	county of res (unless this is only derived from zipcode)
facility code	hospital code
place of death	hospital location (unless facility code-hospital code link makes this redundant)

Create death file

The death file items that I can use for linking are split across the statistical file and the names file. Here, I combine the two files, keeping certificate number and the items I will use in linking.

Listing 1: create death file for linking

```
Steps:
1. read the death file with names
2. merge with standard death file to add dob, age, sex, race,
*/
/*
Step 1. read the death file with names
*/
```

```
data names;
   infile "c:\data\death\deathnames\deathnamesv3.2012" lrecl=241;
   input
      @1   certno     $char10.
      @11  lastname   $char50.
      @61  firstname  $char30.
      @91  middlename $char40.
      @131 suffix     $char4.
      @142 ssnL4      $char4.
      @146 street     $char35.
      @181 city       $char30.
      @213 statecode  $char2.
      ;
run;
/*
Step 2. merge with standard death file
*/
proc sort data=names;
   by certno;
run;
proc sort data=death.dea2012
      out=stats(keep=certno age dob sex cnty_res zipcode dth_date race_wht race_blk
      race_ami race_asi race_chi race_fil race_gua race_haw race_jap race_kor
      race_opi race_oas race_oth race_sam race_vie hisp zipcode facility fac_type);
   by certno;
run;
/*
combine statistical and name files, and recode race fields to match the reduced
set in the CHARS file (which has only white, black, american indian or alaska
native, asian, hawaiian or other Pacific Islander)
*/
data dwnames(drop=sum_race_asi sum_race_haw race_temp1 race_temp2 race_chi race_fil
          race_gua race_jap race_kor race_opi race_oas race_oth race_sam
          race_vie firsttemp lasttemp middlename hisp firsttemp2 lasttemp2);
   length firstname lastname $ 20 miname hispanic $ 1 lastname_sdx firstname_sdx $ 4
          firsttemp2 lasttemp2 $ 25;
   merge stats(rename=(race_asi=race_temp1 race_haw=race_temp2))
         names(rename=(firstname=firsttemp lastname=lasttemp));
   by certno;
   firsttemp2 = compress(firsttemp," ''-_,.&");
   lasttemp2  = compress(lasttemp," ''-_,.&");
   firstname = substr(firsttemp2,1,20);
   lastname  = substr(lasttemp2,1,20);
   miname    = substr(middlename,1,1);
   sum_race_asi = min(1,(race_chi='Y')+(race_fil='Y')+(race_jap='Y')+(race_kor='Y')+
                  (race_oas='Y')+(race_vie='Y')+(race_temp1='Y'));
   sum_race_haw = min(1,(race_gua='Y')+(race_opi='Y')+(race_sam='Y')+(race_temp2='Y'));

   if sum_race_asi = 0 then race_asi = 'N';
   else                     race_asi = 'Y';
   if sum_race_haw = 0 then race_haw = 'N';
   else                     race_haw = 'Y';
   if race_ami in ('') then race_ami = 'U';
   if race_asi in ('') then race_asi = 'U';
   if race_blk in ('') then race_blk = 'U';
   if race_haw in ('') then race_haw = 'U';
   if race_wht in ('') then race_wht = 'U';
   select(hisp);
      when('0')                hispanic = 'N';
      when('1','2','3','4','5') hispanic = 'Y';
      when('','9')             hispanic = 'U';
      end;
   lastname_sdx  = soundex(lastname);
   firstname_sdx = soundex(firstname);

   format dob mmddyy10.;
run;
```

Create CHARS file

Listing 2: Create CHARS file for linking

```
proc format;
   value $stateres
     'AL' = '01'
     'AK' = '02'
     'AZ' = '03'
     'AR' = '04'
     'CA' = '05'
     'CO' = '06'
     'CT' = '07'
     'DE' = '08'
     'DC' = '09'
     'FL' = '10'
     'GA' = '11'
     'HI' = '12'
     'ID' = '13'
     'IL' = '14'
     'IN' = '15'
     'IA' = '16'
     'KS' = '17'
     'KY' = '18'
     'LA' = '19'
     'ME' = '20'
     'MD' = '21'
     'MA' = '22'
     'MI' = '23'
     'MN' = '24'
     'MS' = '25'
     'MO' = '26'
     'MT' = '27'
     'NE' = '28'
     'NV' = '29'
     'NH' = '30'
     'NJ' = '31'
     'NM' = '32'
     'NY' = '33'
     'NC' = '34'
     'ND' = '35'
     'OH' = '36'
     'OK' = '37'
     'OR' = '38'
     'PA' = '39'
     'RI' = '40'
     'SC' = '41'
     'SD' = '42'
     'TN' = '43'
     'TX' = '44'
     'UT' = '45'
     'VT' = '46'
     'VA' = '47'
     'WA' = '48'
     'WV' = '49'
     'WI' = '50'
     'WY' = '51'
     'PR' = '52'
     'VI' = '53'
     'GU' = '54'
     'AS' = '60'
     'MP' = '69'
      ;
run;
data clink1112(keep=seq_no_enc adm_date age country countyres dis_date dob firstname
                ssnL4 hispanic hospital lastname miname race_ami race_asi race_blk
                race_haw race_wht sex statecode status zipcode zipplus4
                lastname_sdx firstname_sdx suffix);
```

```
    length firstname lastname $ 20 suffix $ 4 lastname_sdx firstname_sdx $ 4 statecode $ 2;
    set chars.chr_r2011(rename=(SSN=ssnL4 firstname=firsttemp lastname=lasttemp))
        chars.chr_r2012(rename=(SSN=ssnL4 firstname=firsttemp lastname=lasttemp));
    if race_ami in ('','R') then race_ami = 'U';
    if race_asi in ('','R') then race_asi = 'U';
    if race_blk in ('','R') then race_blk = 'U';
    if race_haw in ('','R') then race_haw = 'U';
    if race_wht in ('','R') then race_wht = 'U';
    if hispanic in ('','R') then hispanic = 'U';
/*
remove the suffixes II, III, IV, V, VI, VII, VIII, ESQ, JR, and SR
from lastnames and place them in a separate suffix field.
Used with UB04 data.
*/
    if _N_ = 1 then do;
     retain __re __reIII;
     pattern = "/( II| III| IV| V| VI| VII| VIII| ESQ|.JR|.SR)$/i";
     __re = prxparse(pattern);
     __reIII = prxparse('/III$/');
    end;
    lasttemp = translate(lasttemp,' ','.,');
    call prxsubstr(__re, TRIM(lasttemp), position, length);
    if position ^= 0 then do;
     suffix    = substr(lasttemp, position + 1, length - 1);
     lasttemp2 = substr(lasttemp, 1, position - 1);
    end;
    else lasttemp2 = lasttemp;

    firstname = compress(firsttemp," ''-_,.&");
    lastname  = compress(lasttemp2," ''-_,.&");
    lastname_sdx  = soundex(lastname);
    firstname_sdx = soundex(firstname);
    statecode = put(stateres,$stateres.);
    if not ('01' le statecode le '69') then statecode = '99';
run;
```

Test birthdate as a blocking field

I will use a SAS program to compute a linkage score for every pair of records in the match between the 2012 death file and 2012 CHARS file. I will evaluate the scores to see if there are any high scores for pairs in which the birthdate does not match. If there are not any such pairs, then birthdate is a good blocking field. I might also evaluate last name in the same way.

Fields I will use, and the points I will give for a matching value are:

item	match	different
age	5	-5
birthdate	20	-20
firstname	10 (2 for soundex match)	-10
lastname	15 (4 for soundex)	-15
middleinit	2	-3
sex	2	-20
zipcode	3	-2
county	3	-5
ssnL4	15	-10
race_ami	5	-5
race_asi	5	-5
race_blk	5	-5
race_haw	5	-5
race_wht	5	-5
hispanic	5	-5
statecode	1	-5
deathdate	10	-10
hospital	5	-10

Listing 3: compute test link scores

```
libname dihd 'c:\data\dihd';

/*
For each record, I will evaluate its similarity with each of the other records
by computing a score using the points described above. In the output dataset,
I will keep records that have a score of at least 0.
Maximum score is 112.
*/
data dihd.link2012;
   set clink1112(rename=(
     age           = c_age
     countyres     = c_cnty_res
     dob           = c_dob
     firstname     = c_firstname
     hispanic      = c_hispanic
     lastname      = c_lastname
     miname        = c_miname
     race_ami      = c_race_ami
     race_asi      = c_race_asi
     race_blk      = c_race_blk
     race_haw      = c_race_haw
     race_wht      = c_race_wht
     sex           = c_sex
     zipcode       = c_zipcode
     firstname_sdx = c_firstname_sdx
     lastname_sdx  = c_lastname_sdx
     ssnl4         = c_ssnl4
     statecode     = c_statecode
     ));
   do i = 1 to 51241;
      set dwnames point=i;
      score =
      (age          = c_age          and age          ne .)*5  +
      (age          ne c_age         )*(-5) +
      (cnty_res     = c_cnty_res     and cnty_res     ne '')*3  +
      (cnty_res     ne c_cnty_res    )*(-5) +
      (dob          = c_dob          and dob          ne .)*20  +
      (dob          ne c_dob         )*(-20) +
      (firstname    = c_firstname    and firstname    ne '')*10  +
      (firstname    ne c_firstname   )*(-10) +
      (hispanic     = c_hispanic     and hispanic     ne '')*5  +
      (hispanic     ne c_hispanic    )*(-5) +
      (lastname     = c_lastname     and lastname     ne '')*15  +
      (lastname     ne c_lastname    )*(-15) +
```

```
      (miname        = c_miname        and miname       ne '')*2  +
      (miname        ne c_miname       )*(-3) +
      (race_ami      = c_race_ami      and race_ami     ne '')*5  +
      (race_ami      ne c_race_ami     )*(-5) +
      (race_asi      = c_race_asi      and race_asi     ne '')*5  +
      (race_asi      ne c_race_asi     )*(-5) +
      (race_blk      = c_race_blk      and race_blk     ne '')*5  +
      (race_blk      ne c_race_blk     )*(-5) +
      (race_haw      = c_race_haw      and race_haw     ne '')*5  +
      (race_haw      ne c_race_haw     )*(-5) +
      (race_wht      = c_race_wht      and race_wht     ne '')*5  +
      (race_wht      ne c_race_wht     )*(-5) +
      (sex           = c_sex           and sex          ne '')*2  +
      (sex           ne c_sex          )*(-20) +
      (zipcode       = c_zipcode       and zipcode      ne '')*3  +
      (zipcode       ne c_zipcode      )*(-2) +
      (firstname_sdx = c_firstname_sdx and firstname_sdx ne '')*2  +
      (firstname_sdx ne c_firstname_sdx)*(-10) +
      (lastname_sdx  = c_lastname_sdx  and lastname_sdx ne '')*4  +
      (lastname_sdx  ne c_lastname_sdx )*(-10) +
      (ssnl4         = c_ssnl4         and ssnl4        ne '')*15  +
      (ssnl4         ne c_ssnl4        )*(-10) +
      (statecode     = c_statecode     and statecode       ne '')*1  +
      (statecode     ne c_statecode    )*(-5) +
      (status = '20' and dth_date = dis_date)*10 +
      (status = '20' and dth_date ne dis_date)*(-10) +
      (status = '20' and facility = substr(hospital,1,3))*5 +
      (status = '20' and facility ne substr(hospital,1,3))*(-10)
      ;

      if score ge 0 then output;
      *output;

      end;
run;

proc print data=dihd.link2012(obs= );
   var score certno firstname c_firstname lastname c_lastname miname c_miname dob c_dob
       ssnL4 c_ssnL4 age c_age sex c_sex cnty_res c_cnty_res statecode c_statecode
       race_ami c_race_ami race_asi c_race_asi race_blk c_race_blk race_haw c_race_haw
       race_wht c_race_wht hispanic c_hispanic;
run;
/*
did any pairs have a high score without birthdate matching?
*/
proc freq data=dihd.link2012;
   where dob ne c_dob;
   tables score;
run;
/*
Answer: yes, there were 550 record pairs where birthdates did not match, but the match score
was 50 or higher.
(I expect around 30,000-60,000 matches, so this is about 1%.)
*/
/*
what about SSN?
*/
proc freq data=dihd.link2012;
   where ssnL4 ne c_ssnL4;
   tables score;
run;
/*
There are more than 2,000 records with different SSNs and high match scores
*/
/*
what about sex?
*/
proc freq data=dihd.link2012;
```

```
   where sex ne c_sex;
   tables score;
run;
/*
There are only 35 records where sex doesn't match and the match score
is above 50.
*/
```

The files are too large to allow for linking in R if we do not block on birthdate, but it seems that blocking on birthdate will cause us to lose about 500 links. (Blocking on SSN loses about 2,000 links; blocking on sex only loses about 30, but does little good.) I will try this: I will link the entire file while blocking on birthdate. I will also create files that contain the death records and the CHARS records from all the pairs for which birthdate did not match, but the matching score was 20 or more, and do the linking routine with them separately, not blocking on birthdate. Then I will combine all the links into one file.

Prepare files for linking

In a preliminary try at linking I saw a pair of records for a baby in which the first and last names did not match (the last names differed on two letters and it looked like one could be a misspelled version of the other, and on the CHARS record the first and middle names were "BABY G"). Race information was also missing on CHARS, so there was liitle to indicate that these records should match each other. But I looked at the original CHARS and death records and saw that the CHARS record showed status 20 (deceased) with same date of death and facility code as the death record. So I conlcuded that these records do match. This example motivated me to make the following changes.

For CHARS records in which status is 20, I will assume the discharge date is the date of death, and the facility code is the facility where death occurred. These should match corresponding fields in the death file. So I will include these fields in the files for linking. In the CHARS linking file, these fields will be blank when status is not 20.

I will prepare switched name fields so that the matching algorithm can compare the first names in the death file to the last names in the CHARS file and vice versa (because I noticed that the names were switched on some CHARS records). I will do this by copying the first and last names in the death file into fields named `deathfirst` and `deathlast` respectively, and copying the first ane last names in the CHARS file into fields named `charsfirst` and `charslast` respectively, and then ordering the fields so that `deathfirst` is compared to `charslast` and `deathlast` is compared to `charsfirst`.

Prepare files to write to R. I convert strings that indicate missing values (such as "" for the race codes, and '9999' for SSN) to blanks so that the linking routines won't think these represent good information.

Listing 4: write death and CHARS files to csv for R

```
/*
the length statements are to ensure fields are in a consistent order
when I read them into R, and the fields are ordered for easiest use
during the classification of the training set.
*/
data dwnames2;
   length certno $ 10 dob 8 firstname $ 20 miname $ 1 lastname $ 20
          suffix $ 4 ssnL4 $ 4 sex $ 1 zipcode $ 5 cnty_res $ 2 facility $ 3
          dth_date 8 hispanic race_wht race_blk race_ami race_asi
          race_haw $ 1 statecode $ 2 deathfirst $ 20 deathlast $ 20;
   set dwnames;
   keep certno cnty_res dob firstname hispanic lastname miname race_ami
    race_asi race_blk race_haw race_wht sex ssnL4 statecode
        zipcode facility dth_date deathfirst deathlast suffix
        ;
   if firstname in ('B','BABY','BABYBOY','BABYGIRL','BOY','GIRL')
      then firstname = '';
   deathfirst = firstname;
   deathlast  = lastname;
   if hispanic = 'U' then hispanic = '';
```

```
   if race_wht = 'U' then race_wht = '';
   if race_blk = 'U' then race_blk = '';
   if race_ami = 'U' then race_ami = '';
   if race_asi = 'U' then race_asi = '';
   if race_haw = 'U' then race_haw = '';
   if sex = 'U' then sex = '';
   if statecode = '99' then statecode = '';
   if zipcode = '99999' then zipcode = '';
   if facility in ('899','999') then facility = '';
   if ssnL4 = '9999' then ssnL4 = '';
   format dth_date mmddyy10.;
run;
proc export data=dwnames2
   outfile = "c:\data\DIHD\death2012.txt"
   dbms = csv
   replace
   ;
run;

data clink2;
   length seq_no_enc $ 10 dob 8 firstname $ 20 miname $ 1 lastname $ 20
          suffix $ 4 ssnL4 $ 4 sex $ 1 zipcode $ 5 countyres $ 2 facility $ 3
          dth_date 8 hispanic race_wht race_blk race_ami race_asi
          race_haw $ 1 statecode $ 2 charslast $ 20 charsfirst $ 20;
   set clink1112;
   if status = '20' then do;
      facility = substr(hospital,1,3);
      dth_date = dis_date;
      end;
   else do;
      facility = '';
      dth_date = .;
      end;
   if firstname in ('B','BABY','BABYBOY','BABYGIRL',
      'BOY','GIRL','BB','BBABY','BABYA','BABYB','BABYBOY',
      'BABYG','BABYGIRL','BABYTWIN','BABYABOY','BABYBGIRL',
      'BABYBOY','BABYBOYA','BABYBOYB','BABYFEMAL','BABYFEMALE',
      'BABYGIRL','BABYGIRLA','BABYGIRLB','BABYMALE','BABYONE',
      'BABYTWO')
      then firstname = '';
   charsfirst = firstname;
   charslast = lastname;
   if hispanic = 'U' then hispanic = '';
   if race_wht = 'U' then race_wht = '';
   if race_blk = 'U' then race_blk = '';
   if race_ami = 'U' then race_ami = '';
   if race_asi = 'U' then race_asi = '';
   if race_haw = 'U' then race_haw = '';
   if sex = 'U' then sex = '';
   if statecode = '99' then statecode = '';
   if zipcode = '99999' then zipcode = '';
   if facility in ('899','999') then facility = '';
   if ssnL4 = '9999' then ssnL4 = '';
   keep seq_no_enc countyres dob firstname hispanic lastname miname race_ami
    race_asi race_blk race_haw race_wht sex ssnL4 statecode
        zipcode facility dth_date charsfirst charslast suffix;
   format dth_date mmddyy10.;
run;
proc export data=clink2
   outfile = "c:\data\DIHD\chars2011_2012.txt"
   dbms = csv
   replace
   ;
run;
```

Perform linking

Now read the files into R.

```
%<<>>=
library(RecordLinkage)
death2012 <- read.csv("../../../data/DIHD/death2012.txt",colClasses=c(rep("character",18)),
                      col.names=c("certno","dob","firstname","miname","lastname","suffix",
                      "ssnL4","sex","zipcode","county","facility","deathdate","hispanic",
                      "race.wht","race.blk","race.ami","race.asi","race.haw","statecode",
                      "death.first","death.last"))
death2012$firstname.sdx <- soundex(death2012$firstname)
death2012$lastname.sdx  <- soundex(death2012$lastname)

chars1112 <- read.csv("../../../data/DIHD/chars2011_2012.txt",colClasses=c(rep("character",18)),
                      col.names=c("seq_no_enc","dob","firstname","miname","lastname","suffix",
                      "ssnL4","sex","zipcode","county","facility","deathdate","hispanic",
                      "race.wht","race.blk","race.ami","race.asi","race.haw","statecode",
                      "chars.last","chars.first"))
chars1112$firstname.sdx <- soundex(chars1112$firstname)
chars1112$lastname.sdx  <- soundex(chars1112$lastname)


#tdeath <- death2012[1:1000,]
#tchars <- chars1112[640000:680000,]

trylcomp <- compare.linkage(tdeath,tchars,blockfld=c(2),exclude=c(1,15,17,18))
trylcomp.sc <- compare.linkage(tdeath,tchars,blockfld=c(2),exclude=c(1,15,17,18),strcmp=c(3,5),
               strcmpfun=levenshteinSim)


# question: can I train a binary comparison dataset and use it to
#  classify a dataset with string metrics?

trylcomp.model <- trainSupv(trylcomp.fsWt.train,method='bagging')
trylcomp.sc.model <- trainSupv(trylcomp.sc.fsWt.train,method='bagging')

trylcomp.result.a <- classifySupv(trylcomp.model,newdata=trylcomp.fsWt)
trylcomp.result.b <- classifySupv(trylcomp.model,newdata=trylcomp.sc.fsWt)
trylcomp.sc.result <- classifySupv(trylcomp.model,newdata=trylcomp.sc.fsWt)

plot(density(trylcomp.result.a$Wdata[trylcomp.result.a$prediction=='N']),xlim=c(-50,130))
lines(density(trylcomp.result.a$Wdata[trylcomp.result.a$prediction=='L']),col=2,lwd=2)

plot(density(trylcomp.result.b$Wdata[trylcomp.result.b$prediction=='N']),xlim=c(-50,130))
lines(density(trylcomp.result.b$Wdata[trylcomp.result.b$prediction=='L']),col=2,lwd=2)

plot(density(trylcomp.sc.result$Wdata[trylcomp.sc.result$prediction=='N']),xlim=c(-50,130))
lines(density(trylcomp.sc.result$Wdata[trylcomp.sc.result$prediction=='L']),col=2,lwd=2)

table(trylcomp.result.a$prediction,trylcomp.result.b$prediction)
table(trylcomp.result.a$prediction,trylcomp.sc.result$prediction)

# result.a and sc.result make exactly the same predictions; result.b
#  gets 2 different, and both those 2 are false matches.


pairs1112 <- compare.linkage(death2012,chars1112,blockfld=c(2),exclude=c(1))

# calculate Fellegi-Sunter weights
pairs1112.fsWt <- fsWeights(pairs1112)

# get a training set
train1112.a <- getMinimalTrain(pairs1112.fsWt,nEx=3)

train1112.a <- editMatch(train1112.a)
```

```
plot(density(pairs1112.fsWt$Wdata,bw=4),col=4,lwd=4)
lines(density(train1112.a$Wdata[train1112.a$pairs$is_match==1]),col=2)
lines(density(train1112.a$Wdata[train1112.a$pairs$is_match==0]),col=1)

model1112.bag <- trainSupv(train1112.a,method='bagging')
result1112.bag <- classifySupv(model1112.bag,newdata=pairs1112.fsWt)

# save the old training set, model, and results
# (these are from before I normalized the names)
save(list=c('train1112.a.old','model1112.bag.old','result1112.bag.old'),file='OldClassifier')


%@
```

A look at the predictive power of the fields suggests that `race.haw` and `race.ami` have little predictive ability. The field `statecode` also doesn't add much.

Using a string similarity metric apparently increases the number of pairs that need to be evaluated for the training set, so I will use it only for first and last names, and not for SSN. After trying that, I found that the number of pairs in the training set increased to a number too high for me to classify (83,000 pairs). I also found that in the small trial I ran, the model that used string comparators classified all the records in the dataset in exactly the same way as the model that did not use string comparators. So I won't use them.

Since I am not using string comparators, I will use all the fields, including those that don't add much.

what I need to do: 1. remove all non-letter characters from first and last names 2. create fields to compare first to last names 3. re-run compare.linkage with string comparators, etc 4. classify the new training set 5. use it to classify the 2012 death records 6. manual review 7. classify the records which had a high score and non-matching birthdate. 8 repeat steps 5-7 for other years.

```
> table(cut(result1112.bag$Wdata[result1112.bag$prediction=='L'],breaks=c(-500,-100,-50,-20,0,20,30,40,50,100,500)))

(-500,-100]  (-100,-50]   (-50,-20]     (-20,0]      (0,20]     (20,30]     (30,40]     (40,50]    (50,100]   (100,500]
          0           0           4          40         275         324         494        1003       42896       44357
> table(cut(result1112.bag$Wdata[result1112.bag$prediction=='N'],breaks=c(-500,-100,-50,-20,0,20,30,40,50,100,500)))

(-500,-100]  (-100,-50]   (-50,-20]     (-20,0]      (0,20]     (20,30]     (30,40]     (40,50]    (50,100]   (100,500]
          0      207213     1642634      102195       12467        1414         161          65          11           0
>
```

I will probably manually review the non-links with weight of 30 or more, and links with weights of 30 or less.

To review links, I subset the RecLinkData like this:

```
manualreview2012 <- result1112.bag[(result1112.bag$prediction=='L'&result1112.bag$Wdata<=30)|
                     (result1112.bag$prediction=='N'&result1112.bag$Wdata>=30)]

manualreview2012 <- editMatch(manualreview2012)

manualreview2012.b <- manualreview2012
for(i in 1:length(manualreview2012$prediction)) {
      manualreview2012.b$prediction[i] <- if(manualreview2012$pairs$is_match[i]==0) 'N' else 'L'
}

predictions.2012a <- result1112.bag$prediction
index.r <- as.numeric(row.names(manualreview2012.b$pairs))
predictions.2012b <- predictions.2012a
predictions.2012b[index.r] <- manualreview2012.b$prediction

# combine death and CHARS row numbers with the predictions
newresults <- cbind(result1112.bag$pairs[,c(1,2)],predictions.2012b)
```

```
# get death certificate numbers and CHARS seq number (seq_no_enc)
deathcerts <- result1112.bag$data1[newresults[,1],1]
charsseq <- result1112.bag$data2[newresults[,2],1]
newresults.b <- data.frame(deathcerts,charsseq,predictions.2012b)
```

Now I get a file of the record pairs which had a high matching score (30 or higher) with non-matching birthdates, and export them to an Excel spreadsheet to conduct a manual review on them. I chose 30 as the cutoff score for manual review because that provides a reasonable number of records for review (about 2,400 for 2012), but I think it includes nearly all the records that have much chance of being classified a true match.

Listing 5: get non-matching birthdate high scorers for manual review

```
libname dihd 'c:\data\dihd';
data review1;
   set dihd.link2012(where=(dob ne c_dob and score ge 30));
run;
proc sort data=review1;
   by score;
run;
data review2(keep=dcert cseq bd fname mi lname ssn sx hosp dd zip county hisp rw
                  rb ram ras rh sc);
   length dcert $ 10 cseq $ 10 bd 8 fname $ 20 mi $ 1 lname $ 20 ssn $ 4 sx $ 1
          hosp $ 3 dd 8 zip $ 5 county $ 2 hisp rw rb ram ras rh $ 1
          sc 8;
   set review1;
   format bd dd mmddyy10.;

   dcert = certno;
   cseq = seq_no_enc;
   bd = dob;
   fname = firstname;
   mi = miname;
   lname = lastname;
   ssn = ssnL4;
   sx = sex;
   hosp = facility;
   dd = dth_date;
   zip = zipcode;
   county = cnty_res;
   hisp = hispanic;
   rw = race_wht;
   rb = race_blk;
   ram = race_ami;
   ras = race_asi;
   rh = race_haw;
   sc = score;
   output;

   bd = c_dob;
   fname = c_firstname;
   mi = c_miname;
   lname = c_lastname;
   ssn = c_ssnL4;
   sx = c_sex;
   if status = 20 or dis_date ge dth_date then do;
      hosp = hospital;
      dd = dis_date;
      end;
   else do;
      hosp = '';
      dd = .;
      end;
   zip = c_zipcode;
   county = c_cnty_res;
```

```
    hisp = c_hispanic;
    rw = c_race_wht;
    rb = c_race_blk;
    ram = c_race_ami;
    ras = c_race_asi;
    rh = c_race_haw;
    sc = .;
    output;

    bd = .;
    fname = '';
    mi = '';
    lname = '';
    ssn = '';
    sx = '';
    hosp = '';
    dd = .;
    zip = '';
    county = '';
    hisp = '';
    rw = '';
    rb = '';
    ram = '';
    ras = '';
    rh = '';
    sc = .;
    output;
run;

proc export data=review2
    outfile = "c:\user\projects\Death-CHARSlink\manreview2012.xls"
    dbms = excel5
    replace
    ;
run;
/*
read the reviewed links
*/
proc import out=review3
    file = "c:\user\projects\Death-CHARSlink\manreview2012_done.xls"
    dbms = excel5
    ;
run;
```

Notes for future years:

1. convert '9999' in ssn to missing so it doesn't add to the score
2. subtract from the score if the discharge date is more than one day past the death date.
3. compare the elements (day, month, year) of the birth date and add to the score if some of them are the same.

Now I need to combine the links from three sources: the machine learning results, the manual review of those results, and the manual coding of the records on which birthdate didn't match. After combining those links, I need to check whether there are any hospitalization records linked to more than one death record, and if so, adjudicate those links manually. Then I can create the final linked file.

```
#create file containing only the linked pairs
links2012 <- newresults.b[newresults.b$predictions.2012b=='L',]

write.csv(links2012,file="c:/data/dihd/links2012.csv",row.names=F)
```

Listing 6: create final linked file for 2012

```
libname dihd 'c:\data\dihd';
```

```
proc import out=links0
    file = "c:\data\dihd\links2012.csv"
    dbms = csv
    replace
    ;
run;
data links1(keep=certno seq_no_enc predict);
    length certno seq_no_enc $ 10 predict $ 1;
    set links0;
    certno = substr(deathcerts,1,10);
    seq_no_enc = substr(charsseq,1,10);
    predict = substr(predictions_2012b,1,1);
run;
/*
find the CHARS records that linked to more than one death certificate
(there are 7 CHARS records that each linked to 2 death certs, and 3
    that each linked to 3 death certs)
*/
proc freq data=links1 noprint;
    tables seq_no_enc/out=charslist;
run;
data mults1(drop=percent);
    set charslist(where=(count ge 2));
run;
proc sort data=links1;
    by seq_no_enc;
run;
data mults2;
    merge links1 mults1(in=inmult);
    by seq_no_enc;
    if inmult;
run;
/*
I'll guess that all these pairs are in the dataset with hig scores,
    and I will get the detailed information from there.
*/
proc sort data=dihd.link2012;
    by certno seq_no_enc;
run;
proc sort data=mults2;
    by certno seq_no_enc;
run;

data mults3;
    merge dihd.link2012 mults2(in=inmult);
    by certno seq_no_enc;
    if inmult;
run;
proc print data=mults3;
run;
/*
I code the pairs by hand and enter the data here
*/
data mults4;
    input @1 certno $char10. @12 seq_no_enc $char10. @23 link $char1.;
datalines;
2012010713 2012107470 N
2012010713 2012287081 N
2012010717 2012107470 N
2012010717 2012287081 N
2012056614 2012253552 N
2012056614 2012579510 N
2012056615 2012253552 N
2012056615 2012579510 N
2012058625 2012059910 L
2012058625 2012457731 N
2012058626 2012059910 N
```

```
2012058626 2012457731 L
2012063085 2012087235 N
2012063085 2012457771 N
2012063085 2012551638 N
2012063086 2012087235 N
2012063086 2012457771 N
2012063086 2012551638 N
2012063087 2012087235 N
2012063087 2012457771 N
2012063087 2012551638 N
2012090096 2011083190 L
2012091639 2011083190 L
;;
run;
/*
I found that death certificates 2012090096 and 2012091639 seem to be
   for the same person.
*/
proc sort data=links1;
   by certno seq_no_enc;
run;
data links2(keep=certno seq_no_enc);
   merge links1 mults4;
   by certno seq_no_enc;
   if link = '' then match = predict;
   else              match = link;
   if match = 'L' then output;
run;

/*
read in the reviewed links for pairs which had high scores but
   non-matching birthdates
*/
proc import out=review3
   file = "c:\user\projects\Death-CHARSlink\manreview2012_done.xls"
   dbms = excel5
   ;
run;
data mlinks1(keep=certno seq_no_enc sc link);
   length certno seq_no_enc $ 10;
   retain i 0;
   set review3;
   certno = substr(dcert,1,10);
   seq_no_enc = substr(cseq,1,10);
   i+1;
   if i = 1 then output;
   if i = 3 then i = 0;
run;
proc freq data=mlinks1;
   tables sc*link/norow nocol nopercent;
run;
/*
this table shows the strong relation between score and link status

            The SAS System
09:42 Monday, June 2, 2014   23

          The FREQ Procedure

         Table of SC by LINK

 SC(SC)      LINK(LINK)

 Frequency|        0|        1|   Total
----------+---------+---------+
       30 |      62 |      10 |      72
----------+---------+---------+
       31 |       5 |       0 |       5
```

32	858	10	868
33	0	2	2
34	125	0	125
35	4	9	13
36	1	0	1
37	197	34	231
38	0	3	3
39	0	7	7
40	4	2	6
41	0	3	3
42	16	36	52
43	0	1	1
44	1	7	8
45	0	3	3
46	0	1	1
47	1	68	69
48	0	1	1
49	0	30	30
50	0	4	4
52	1	84	85
53	0	4	4
54	0	3	3
55	0	17	17
57	0	102	102
59	0	12	12
60	0	30	30
62	0	263	263
64	0	3	3
65	0	6	6
67	0	61	61
68	0	1	1
70	0	33	33
72	0	214	214
74	0	3	3

```
+---------+---------+---------+
|      75 |       0 |       5 |       5
+---------+---------+---------+
|      77 |       0 |      27 |      27
+---------+---------+---------+
|      82 |       0 |      15 |      15
+---------+---------+---------+
|      85 |       0 |       5 |       5
+---------+---------+---------+
|      87 |       0 |      49 |      49
+---------+---------+---------+
 Total         1275      1168      2443
*/
data mlinks2(keep=certno seq_no_enc);
   set mlinks1(where=(link=1));
run;
data dihd.finallink2012;
   set links2 mlinks2;
run;

proc freq data=dihd.finallink2012 noprint;
   tables certno/out=dcertlist;
run;
/*
I found that 35,993 death certificates (subtracting one copy of the
   duplicate I found) linked to 90,554 hospital records.
*/
```

DIHD file for 2011

Listing 7: create death file for linking

```
Steps:
1. read the death file with names
2. merge with standard death file to add dob, age, sex, race,
*/
/*
Step 1. read the death file with names
*/
data names;
   infile "c:\data\death\deathnames\deathnamesv3.2011" lrecl=241;
   input
      @1   certno      $char10.
      @11  lastname    $char50.
      @61  firstname   $char30.
      @91  middlename  $char40.
      @131 suffix      $char4.
      @158 ssnL4       $char4.
      @162 street      $char35.
      @197 city        $char30.
      @236 statecode   $char2.
      ;
run;
/*
Step 2. merge with standard death file
*/
proc sort data=names;
   by certno;
run;
proc sort data=death.dea2011
      out=stats(keep=certno age dob sex cnty_res zipcode dth_date race_wht race_blk
      race_ami race_asi race_chi race_fil race_gua race_haw race_jap race_kor
      race_opi race_oas race_oth race_sam race_vie hisp zipcode facility fac_type);
   by certno;
run;
/*
```

```
combine statistical and name files, and recode race fields to match the reduced
set in the CHARS file (which has only white, black, american indian or alaska
native, asian, hawaiian or other Pacific Islander)
*/
data dwnames(drop=sum_race_asi sum_race_haw race_temp1 race_temp2 race_chi race_fil
          race_gua race_jap race_kor race_opi race_oas race_oth race_sam
          race_vie firsttemp lasttemp middlename hisp firsttemp2 lasttemp2);
   length firstname lastname $ 20 miname hispanic $ 1 lastname_sdx firstname_sdx $ 4
          firsttemp2 lasttemp2 $ 25;
   merge stats(rename=(race_asi=race_temp1 race_haw=race_temp2))
         names(rename=(firstname=firsttemp lastname=lasttemp));
   by certno;
   firsttemp2 = compress(firsttemp," ''-_,.&");
   lasttemp2  = compress(lasttemp," ''-_,.&");
   firstname = substr(firsttemp2,1,20);
   lastname  = substr(lasttemp2,1,20);
   miname    = substr(middlename,1,1);
   sum_race_asi = min(1,(race_chi='Y')+(race_fil='Y')+(race_jap='Y')+(race_kor='Y')+
                  (race_oas='Y')+(race_vie='Y')+(race_temp1='Y'));
   sum_race_haw = min(1,(race_gua='Y')+(race_opi='Y')+(race_sam='Y')+(race_temp2='Y'));

   if sum_race_asi = 0 then race_asi = 'N';
   else                     race_asi = 'Y';
   if sum_race_haw = 0 then race_haw = 'N';
   else                     race_haw = 'Y';
   if race_ami in ('') then race_ami = 'U';
   if race_asi in ('') then race_asi = 'U';
   if race_blk in ('') then race_blk = 'U';
   if race_haw in ('') then race_haw = 'U';
   if race_wht in ('') then race_wht = 'U';
   select(hisp);
      when('0')                 hispanic = 'N';
      when('1','2','3','4','5') hispanic = 'Y';
      when('','9')              hispanic = 'U';
      end;
   lastname_sdx  = soundex(lastname);
   firstname_sdx = soundex(firstname);

   format dob mmddyy10.;
run;
```

Listing 8: Create CHARS file for linking

```
proc format;
   value $stateres
     'AL' = '01'
     'AK' = '02'
     'AZ' = '03'
     'AR' = '04'
     'CA' = '05'
     'CO' = '06'
     'CT' = '07'
     'DE' = '08'
     'DC' = '09'
     'FL' = '10'
     'GA' = '11'
     'HI' = '12'
     'ID' = '13'
     'IL' = '14'
     'IN' = '15'
     'IA' = '16'
     'KS' = '17'
     'KY' = '18'
     'LA' = '19'
     'ME' = '20'
     'MD' = '21'
     'MA' = '22'
     'MI' = '23'
```

```
      'MN' = '24'
      'MS' = '25'
      'MO' = '26'
      'MT' = '27'
      'NE' = '28'
      'NV' = '29'
      'NH' = '30'
      'NJ' = '31'
      'NM' = '32'
      'NY' = '33'
      'NC' = '34'
      'ND' = '35'
      'OH' = '36'
      'OK' = '37'
      'OR' = '38'
      'PA' = '39'
      'RI' = '40'
      'SC' = '41'
      'SD' = '42'
      'TN' = '43'
      'TX' = '44'
      'UT' = '45'
      'VT' = '46'
      'VA' = '47'
      'WA' = '48'
      'WV' = '49'
      'WI' = '50'
      'WY' = '51'
      'PR' = '52'
      'VI' = '53'
      'GU' = '54'
      'AS' = '60'
      'MP' = '69'
       ;
run;
data clink1011(keep=seq_no_enc adm_date age country countyres dis_date dob firstname
                ssnL4 hispanic hospital lastname miname race_ami race_asi race_blk
                race_haw race_wht sex statecode status zipcode zipplus4
                lastname_sdx firstname_sdx suffix);
   length firstname lastname $ 20 suffix $ 4 lastname_sdx firstname_sdx $ 4 statecode $ 2;
   set chars.chr_r2010(rename=(SSN=ssnL4 firstname=firsttemp lastname=lasttemp))
       chars.chr_r2011(rename=(SSN=ssnL4 firstname=firsttemp lastname=lasttemp));
   if race_ami in ('','R') then race_ami = 'U';
   if race_asi in ('','R') then race_asi = 'U';
   if race_blk in ('','R') then race_blk = 'U';
   if race_haw in ('','R') then race_haw = 'U';
   if race_wht in ('','R') then race_wht = 'U';
   if hispanic in ('','R') then hispanic = 'U';
/*
remove the suffixes II, III, IV, V, VI, VII, VIII, ESQ, JR, and SR
from lastnames and place them in a separate suffix field.
Used with UB04 data.
*/
   if _N_ = 1 then do;
    retain __re __reIII;
    pattern = "/( II| III| IV| V| VI| VII| VIII| ESQ|.JR|.SR)$/i";
    __re = prxparse(pattern);
    __reIII = prxparse('/III$/');
   end;
   lasttemp = translate(lasttemp,' ','.,');
   call prxsubstr(__re, TRIM(lasttemp), position, length);
   if position ^= 0 then do;
    suffix    = substr(lasttemp, position + 1, length - 1);
    lasttemp2 = substr(lasttemp, 1, position - 1);
   end;
   else lasttemp2 = lasttemp;

   firstname = compress(firsttemp," ''-_,.&");
```

```
   lastname  = compress(lasttemp2," ''-_,.&");
   lastname_sdx  = soundex(lastname);
   firstname_sdx = soundex(firstname);
*   statecode = put(stateres,$stateres.);
   statecode = stateres;
*   if not ('01' le statecode le '69') then statecode = '99';
   if statecode = 'XX' then statecode = '';
run;
```

Listing 9: compute test link scores

```
libname dihd 'c:\data\dihd';

/*
For each record, I will evaluate its similarity with each of the other records
by computing a score using the points described above. In the output dataset,
I will keep records that have a score of at least 0.
Maximum score is 112.
*/
data dihd.link2011;
   set clink1011(rename=(
     age           = c_age
     countyres     = c_cnty_res
     dob           = c_dob
     firstname     = c_firstname
     hispanic      = c_hispanic
     lastname      = c_lastname
     miname        = c_miname
     race_ami      = c_race_ami
     race_asi      = c_race_asi
     race_blk      = c_race_blk
     race_haw      = c_race_haw
     race_wht      = c_race_wht
     sex           = c_sex
     zipcode       = c_zipcode
     firstname_sdx = c_firstname_sdx
     lastname_sdx  = c_lastname_sdx
     ssnl4         = c_ssnl4
     statecode     = c_statecode
     ));
   do i = 1 to 50589;
      set dwnames point=i;
      score =
      (age           = c_age           and age           ne .)*5  +
      (age           ne c_age          )*(-5) +
      (cnty_res      = c_cnty_res      and cnty_res      ne '')*3  +
      (cnty_res      ne c_cnty_res     )*(-5) +
      (dob           = c_dob           and dob           ne .)*20  +
      (dob           ne c_dob          )*(-20) +
      (firstname     = c_firstname     and firstname     ne '')*10  +
      (firstname     ne c_firstname    )*(-10) +
      (hispanic      = c_hispanic      and hispanic      ne '')*5  +
      (hispanic      ne c_hispanic     )*(-5) +
      (lastname      = c_lastname      and lastname      ne '')*15  +
      (lastname      ne c_lastname     )*(-15) +
      (miname        = c_miname        and miname        ne '')*2  +
      (miname        ne c_miname       )*(-3) +
      (race_ami      = c_race_ami      and race_ami      ne '')*5  +
      (race_ami      ne c_race_ami     )*(-5) +
      (race_asi      = c_race_asi      and race_asi      ne '')*5  +
      (race_asi      ne c_race_asi     )*(-5) +
      (race_blk      = c_race_blk      and race_blk      ne '')*5  +
      (race_blk      ne c_race_blk     )*(-5) +
      (race_haw      = c_race_haw      and race_haw      ne '')*5  +
      (race_haw      ne c_race_haw     )*(-5) +
      (race_wht      = c_race_wht      and race_wht      ne '')*5  +
      (race_wht      ne c_race_wht     )*(-5) +
      (sex           = c_sex           and sex           ne '')*2  +
```

```
      (sex           ne c_sex          )*(-20) +
      (zipcode       = c_zipcode       and zipcode      ne '')*3  +
      (zipcode       ne c_zipcode      )*(-2) +
      (firstname_sdx = c_firstname_sdx and firstname_sdx ne '')*2  +
      (firstname_sdx ne c_firstname_sdx)*(-10) +
      (lastname_sdx  = c_lastname_sdx  and lastname_sdx ne '')*4  +
      (lastname_sdx  ne c_lastname_sdx )*(-10) +
      (ssnl4         = c_ssnl4         and ssnl4        ne '')*15  +
      (ssnl4         ne c_ssnl4        )*(-10) +
      (statecode     = c_statecode     and statecode       ne '')*1  +
      (statecode     ne c_statecode    )*(-5) +
      (status = '20' and dth_date = dis_date)*10 +
      (status = '20' and dth_date ne dis_date)*(-10) +
      (status = '20' and facility = substr(hospital,1,3))*5 +
      (status = '20' and facility ne substr(hospital,1,3))*(-10)
      ;

      if score ge 0 then output;
      *output;

      end;
run;

proc print data=dihd.link2011(obs= );
   var score certno firstname c_firstname lastname c_lastname miname c_miname dob c_dob
       ssnL4 c_ssnL4 age c_age sex c_sex cnty_res c_cnty_res statecode c_statecode
       race_ami c_race_ami race_asi c_race_asi race_blk c_race_blk race_haw c_race_haw
       race_wht c_race_wht hispanic c_hispanic;
run;
/*
did any pairs have a high score without birthdate matching?
*/
proc freq data=dihd.link2011;
   where dob ne c_dob;
   tables score;
run;
```

Listing 10: write death and CHARS files to csv for R

```
/*
the length statements are to ensure fields are in a consistent order
when I read them into R, and the fields are ordered for easiest use
during the classification of the training set.
I delete the records that have no names or SSN (typically these are
deaths that occurred out-of-state).
*/
data dwnames2;
   length certno $ 10 dob 8 firstname $ 20 miname $ 1 lastname $ 20
          suffix $ 4 ssnL4 $ 4 sex $ 1 zipcode $ 5 cnty_res $ 2 facility $ 3
          dth_date 8 hispanic race_wht race_blk race_ami race_asi
          race_haw $ 1 statecode $ 2 deathfirst $ 20 deathlast $ 20;
   set dwnames;
   keep certno cnty_res dob firstname hispanic lastname miname race_ami
    race_asi race_blk race_haw race_wht sex ssnL4 statecode
        zipcode facility dth_date deathfirst deathlast suffix
        ;
   if firstname in ('B','BABY','BABYBOY','BABYGIRL','BOY','GIRL')
      then firstname = '';
   deathfirst = firstname;
   deathlast  = lastname;
   if firstname = '' and lastname = '' then delete;
   if hispanic = 'U' then hispanic = '';
   if race_wht = 'U' then race_wht = '';
   if race_blk = 'U' then race_blk = '';
   if race_ami = 'U' then race_ami = '';
   if race_asi = 'U' then race_asi = '';
   if race_haw = 'U' then race_haw = '';
   if sex = 'U' then sex = '';
```

```
    if statecode = '99' then statecode = '';
    if zipcode = '99999' then zipcode = '';
    if facility in ('899','999') then facility = '';
    if ssnL4 = '9999' then ssnL4 = '';
    format dth_date mmddyy10.;
run;
proc export data=dwnames2
    outfile = "c:\data\DIHD\death2011.txt"
    dbms = csv
    replace
    ;
run;

data clink2;
    length seq_no_enc $ 10 dob 8 firstname $ 20 miname $ 1 lastname $ 20
           suffix $ 4 ssnL4 $ 4 sex $ 1 zipcode $ 5 countyres $ 2 facility $ 3
           dth_date 8 hispanic race_wht race_blk race_ami race_asi
           race_haw $ 1 statecode $ 2 charslast $ 20 charsfirst $ 20;
    set clink1011;
    if status = '20' then do;
       facility = substr(hospital,1,3);
       dth_date = dis_date;
       end;
    else do;
       facility = '';
       dth_date = .;
       end;
    if firstname in ('B','BABY','BABYBOY','BABYGIRL',
       'BOY','GIRL','BB','BBABY','BABYA','BABYB','BABYBOY',
       'BABYG','BABYGIRL','BABYTWIN','BABYABOY','BABYBGIRL',
       'BABYBOY','BABYBOYA','BABYBOYB','BABYFEMAL','BABYFEMALE',
       'BABYGIRL','BABYGIRLA','BABYGIRLB','BABYMALE','BABYONE',
       'BABYTWO')
       then firstname = '';
    charsfirst = firstname;
    charslast = lastname;
    if hispanic = 'U' then hispanic = '';
    if race_wht = 'U' then race_wht = '';
    if race_blk = 'U' then race_blk = '';
    if race_ami = 'U' then race_ami = '';
    if race_asi = 'U' then race_asi = '';
    if race_haw = 'U' then race_haw = '';
    if sex = 'U' then sex = '';
    if statecode = '99' then statecode = '';
    if zipcode = '99999' then zipcode = '';
    if facility in ('899','999') then facility = '';
    if ssnL4 = '9999' then ssnL4 = '';
    keep seq_no_enc countyres dob firstname hispanic lastname miname race_ami
     race_asi race_blk race_haw race_wht sex ssnL4 statecode
         zipcode facility dth_date charsfirst charslast suffix;
    format dth_date mmddyy10.;
run;
proc export data=clink2
    outfile = "c:\data\DIHD\chars2010_2011.txt"
    dbms = csv
    replace
    ;
run;
```

```
%<<>>=
library(RecordLinkage)

# save the previous training set, model, and results
save(list=c('pairs1112.fsWt','train1112.a','model1112.bag','result1112.bag',
        'manualreview2012.b'),file='Classifier2012')


death2011 <- read.csv("../../../data/DIHD/death2011.txt",colClasses=c(rep("character",18)),
```

```
                    col.names=c("certno","dob","firstname","miname","lastname","suffix",
                    "ssnL4","sex","zipcode","county","facility","deathdate","hispanic",
                    "race.wht","race.blk","race.ami","race.asi","race.haw","statecode",
                    "death.first","death.last"))
death2011$firstname.sdx <- soundex(death2011$firstname)
death2011$lastname.sdx  <- soundex(death2011$lastname)

chars1011 <- read.csv("../../../data/DIHD/chars2010_2011.txt",colClasses=c(rep("character",18)),
                    col.names=c("seq_no_enc","dob","firstname","miname","lastname","suffix",
                    "ssnL4","sex","zipcode","county","facility","deathdate","hispanic",
                    "race.wht","race.blk","race.ami","race.asi","race.haw","statecode",
                    "chars.last","chars.first"))
chars1011$firstname.sdx <- soundex(chars1011$firstname)
chars1011$lastname.sdx  <- soundex(chars1011$lastname)

pairs2011 <- compare.linkage(death2011,chars1011,blockfld=c(2),exclude=c(1))

# calculate Fellegi-Sunter weights
pairs2011.fsWt <- fsWeights(pairs2011)

# get a training set
train2011.a <- getMinimalTrain(pairs2011.fsWt,nEx=3)

train2011.a <- editMatch(train2011.a)

model2011.bag <- trainSupv(train2011.a,method='bagging')
result2011.bag <- classifySupv(model2011.bag,newdata=pairs2011.fsWt)

manualreview2011 <- result2011.bag[(result2011.bag$prediction=='L'&result2011.bag$Wdata<=35)|
                    (result2011.bag$prediction=='N'&result2011.bag$Wdata>=20)]

manualreview2011 <- editMatch(manualreview2011)

%@
```

This shows the relationships between the weight, the machine prediction, and the manual classification on the records that I manually reviewed:

```
> with(manualreview2011[manualreview2011$Wdata<0],table(pairs$is_match,prediction))
   prediction
     N  P  L
  0  0  0  4
  1  0  0 33
> with(manualreview2011[manualreview2011$Wdata>=0&manualreview2011$Wdata<10],table(pairs$is_match,prediction))
   prediction
     N  P  L
  0  0  0 19
  1  0  0 91
> with(manualreview2011[manualreview2011$Wdata>=10&manualreview2011$Wdata<20],table(pairs$is_match,prediction))
   prediction
      N   P   L
  0   0   0  36
  1   0   0 169
> with(manualreview2011[manualreview2011$Wdata>=20&manualreview2011$Wdata<30],table(pairs$is_match,prediction))
   prediction
       N    P    L
  0 1199    0   19
  1   27    0  331
> with(manualreview2011[manualreview2011$Wdata>=30&manualreview2011$Wdata<40],table(pairs$is_match,prediction))
   prediction
      N   P   L
  0 114   0   0
```

```
  1  17   0 220
> with(manualreview2011[manualreview2011$Wdata>=40&manualreview2011$Wdata<50],table(pairs$is_match,prediction))
   prediction
     N  P  L
  0 35  0  0
  1 27  0  0
> with(manualreview2011[manualreview2011$Wdata>=50&manualreview2011$Wdata<60],table(pairs$is_match,prediction))
   prediction
     N  P  L
  1 29  0  0
> with(manualreview2011[manualreview2011$Wdata>=60],table(pairs$is_match,prediction))
   prediction
    N P L
  1 5 0 0
```

In tabular form:

When the machine predicted a link (I manually reviewed all pairs where the weight was 35 or less and the machine predicted a link):

weight	match	not	% not match
< 0	33	4	11
0–10	91	19	17
10–20	169	36	18
20–30	331	19	5
30–35	220	0	0

When the machine predicted a pair was not a link (I manually reviewed all pairs where the weight was 20 or more and the machine predicted the pair was not a link):

weight	match	not	% match
20–30	27	1199	2
30–40	17	114	13
40–50	27	35	44
50–60	29	0	100
60 +	5	0	100

So it looks like I should continue to manually review pairs that satisfy one of these two conditions:

- weight is 30 or lower and the machine predicts a link
- weight is 30 or higher and the machine predicts not a link

For 2011, following these guidelines would have meant doing manual review on 929 pairs, and changing the classification of 156 of them (17%).

```
%<<>>=
library(RecordLinkage)

manualreview2011.b <- manualreview2011
for(i in 1:length(manualreview2011$prediction)) {
      manualreview2011.b$prediction[i] <- if(manualreview2011$pairs$is_match[i]==0) 'N' else 'L'
}

predictions.2011a <- result2011.bag$prediction
index.r <- as.numeric(row.names(manualreview2011.b$pairs))
predictions.2011b <- predictions.2011a
predictions.2011b[index.r] <- manualreview2011.b$prediction

# combine death and CHARS row numbers with the predictions
newresults2011 <- cbind(result2011.bag$pairs[,c(1,2)],predictions.2011b)
```

```
# get death certificate numbers and CHARS seq number (seq_no_enc)
deathcerts <- result2011.bag$data1[newresults2011[,1],1]
charsseq <- result2011.bag$data2[newresults2011[,2],1]
newresults2011.b <- data.frame(deathcerts,charsseq,predictions.2011b)

%@
```

Manual review of records with non-matching birthdates

Now I get a file of the record pairs which had a high matching score (30 or higher) with non-matching birthdates, and export them to an Excel spreadsheet to conduct a manual review on them. I chose 30 as the cutoff score for manual review because that provides a reasonable number of records for review (about 2,400 for 2011), but I think it includes nearly all the records that have much chance of being classified a true match.

Listing 11: get non-matching birthdate high scorers for manual review

```
libname dihd 'c:\data\dihd';
data review1;
   set dihd.link2011(where=(dob ne c_dob and score ge 30));
run;
proc sort data=review1;
   by score;
run;
data review2(keep=dcert cseq bd fname mi lname ssn sx hosp dd zip county hisp rw
                  rb ram ras rh sc);
   length dcert $ 10 cseq $ 10 bd 8 fname $ 20 mi $ 1 lname $ 20 ssn $ 4 sx $ 1
          hosp $ 3 dd 8 zip $ 5 county $ 2 hisp rw rb ram ras rh $ 1
          sc 8;
   set review1;
   format bd dd mmddyy10.;

   dcert = certno;
   cseq = seq_no_enc;
   bd = dob;
   fname = firstname;
   mi = miname;
   lname = lastname;
   ssn = ssnL4;
   sx = sex;
   hosp = facility;
   dd = dth_date;
   zip = zipcode;
   county = cnty_res;
   hisp = hispanic;
   rw = race_wht;
   rb = race_blk;
   ram = race_ami;
   ras = race_asi;
   rh = race_haw;
   sc = score;
   output;

   bd = c_dob;
   fname = c_firstname;
   mi = c_miname;
   lname = c_lastname;
   ssn = c_ssnL4;
   sx = c_sex;
   if status = 20 or dis_date ge dth_date then do;
      hosp = hospital;
      dd = dis_date;
      end;
   else do;
      hosp = '';
```

```
      dd = .;
      end;
   zip = c_zipcode;
   county = c_cnty_res;
   hisp = c_hispanic;
   rw = c_race_wht;
   rb = c_race_blk;
   ram = c_race_ami;
   ras = c_race_asi;
   rh = c_race_haw;
   sc = .;
   output;

   bd = .;
   fname = '';
   mi = '';
   lname = '';
   ssn = '';
   sx = '';
   hosp = '';
   dd = .;
   zip = '';
   county = '';
   hisp = '';
   rw = '';
   rb = '';
   ram = '';
   ras = '';
   rh = '';
   sc = .;
   output;
run;

proc export data=review2
   outfile = "c:\user\projects\Death-CHARSlink\manreview2011.xls"
   dbms = excel5
   replace
   ;
run;
/*
read the reviewed links
*/
proc import out=review3
   file = "c:\user\projects\Death-CHARSlink\manreview2011_done.xls"
   dbms = excel5
   ;
run;
```

Now I need to combine the links from three sources: the machine learning results, the manual review of those results, and the manual coding of the records on which birthdate didn't match. After combining those links, I need to check whether there are any hospitalization records linked to more than one death record, and if so, adjudicate those links manually. Then I can create the final linked file.

```
#create file containing only the linked pairs
links2011 <- newresults2011.b[newresults2011.b$predictions.2011b=='L',]

write.csv(links2011,file="c:/data/dihd/links2011.csv",row.names=F)
```

Listing 12: create final linked file for 2011

```
libname dihd 'c:\data\dihd';

proc import out=links0
   file = "c:\data\dihd\links2011.csv"
   dbms = csv
```

```
   replace
   ;
run;
data links1(keep=certno seq_no_enc predict);
   length certno seq_no_enc $ 10 predict $ 1;
   set links0;
   certno = substr(deathcerts,1,10);
   seq_no_enc = substr(charsseq,1,10);
   predict = substr(predictions_2011b,1,1);
run;
/*
find the CHARS records that linked to more than one death certificate
(there are 7 CHARS records that each linked to 2 death certs)
*/
proc freq data=links1 noprint;
   tables seq_no_enc/out=charslist;
run;
data mults1(drop=percent);
   set charslist(where=(count ge 2));
run;
proc sort data=links1;
   by seq_no_enc;
run;
data mults2;
   merge links1 mults1(in=inmult);
   by seq_no_enc;
   if inmult;
run;
/*
I'll guess that all these pairs are in the dataset with high scores,
   and I will get the detailed information from there.
*/
proc sort data=dihd.link2011;
   by certno seq_no_enc;
run;
proc sort data=mults2;
   by certno seq_no_enc;
run;

data mults3;
   merge dihd.link2011 mults2(in=inmult);
   by certno seq_no_enc;
   if inmult;
run;
proc print data=mults3;
run;
/*
I code the pairs by hand and enter the data here
*/
data mults4;
   input @1 certno $char10. @12 seq_no_enc $char10. @23 link $char1.;
datalines;
2011050070 2010130586 N
2011050070 2011112553 N
2011050070 2011212287 N
2011050070 2011385211 N
2011050070 2011445277 N
2011050070 2011547660 N
2011050070 2011612676 N
2011057314 2010130586 L
2011057314 2011112553 L
2011057314 2011212287 L
2011057314 2011385211 L
2011057314 2011445277 L
2011057314 2011547660 L
2011057314 2011612676 L
;;
run;
```

```
proc sort data=links1;
   by certno seq_no_enc;
run;
data links2(keep=certno seq_no_enc);
   merge links1 mults4;
   by certno seq_no_enc;
   if link = '' then match = predict;
   else              match = link;
   if match = 'L' then output;
run;

/*
read in the reviewed links for pairs which had high scores but
   non-matching birthdates
*/
proc import out=review3
   file = "c:\user\projects\Death-CHARSlink\manreview2011_done.xls"
   dbms = excel5
   ;
run;
data mlinks1(keep=certno seq_no_enc sc link);
   length certno seq_no_enc $ 10;
   retain i 0;
   set review3;
   certno = substr(dcert,1,10);
   seq_no_enc = substr(cseq,1,10);
   i+1;
   if i = 1 then output;
   if i = 3 then i = 0;
run;
proc freq data=mlinks1;
   tables sc*link/norow nocol nopercent;
run;
/*
this table shows the strong relation between score and link status
```

The SAS System 12:04 Friday, July 11, 2014 122

The FREQ Procedure

Table of SC by LINK

SC(SC)	LINK(LINK)		
Frequency	0	1	Total
30	36	14	50
32	803	13	816
33	1	1	2
34	99	4	103
35	5	6	11
37	178	20	198
38	1	0	1
39	0	6	6
40	0	4	4
42	11	38	49
43	0	6	6
44	6	7	13

```
+--------+--------+--------+
|     45 |      0 |      7 |       7
+--------+--------+--------+
|     47 |      3 |     40 |      43
+--------+--------+--------+
|     48 |      0 |      1 |       1
+--------+--------+--------+
|     49 |      4 |     25 |      29
+--------+--------+--------+
|     50 |      0 |     18 |      18
+--------+--------+--------+
|     52 |      1 |     92 |      93
+--------+--------+--------+
|     53 |      0 |      6 |       6
+--------+--------+--------+
|     54 |      0 |      3 |       3
+--------+--------+--------+
|     55 |      0 |      4 |       4
+--------+--------+--------+
|     57 |      0 |    135 |     135
+--------+--------+--------+
|     59 |      0 |     14 |      14
+--------+--------+--------+
|     60 |      0 |     12 |      12
+--------+--------+--------+
|     62 |      0 |    271 |     271
+--------+--------+--------+
|     64 |      0 |      1 |       1
+--------+--------+--------+
|     65 |      0 |     13 |      13
+--------+--------+--------+
|     67 |      0 |     95 |      95
+--------+--------+--------+
|     68 |      0 |      3 |       3
+--------+--------+--------+
|     69 |      0 |      1 |       1
+--------+--------+--------+
|     70 |      0 |     13 |      13
+--------+--------+--------+
|     72 |      0 |    265 |     265
+--------+--------+--------+
|     74 |      0 |      1 |       1
+--------+--------+--------+
|     75 |      0 |      3 |       3
+--------+--------+--------+
|     77 |      0 |     32 |      32
+--------+--------+--------+
|     80 |      0 |      1 |       1
+--------+--------+--------+
|     82 |      0 |     21 |      21
+--------+--------+--------+
|     85 |      0 |      4 |       4
+--------+--------+--------+
|     87 |      0 |     49 |      49
+--------+--------+--------+
 Total        1148     1249      2397
*/
data mlinks2(keep=certno seq_no_enc);
   set mlinks1(where=(link=1));
run;
data dihd.finallink2011;
   set links2 mlinks2;
run;

proc freq data=dihd.finallink2011 noprint;
   tables certno/out=dcertlist;
run;
/*
```

```
I found that 35,736 death certificates linked to 90,371
hospital records.
*/
```

DIHD file for 2010

Listing 13: create death file for linking

```
/*
Step 1. read the death file with names
*/
libname death 'c:\data\death';
libname chars 'c:\data\chars';

data names;
   infile "c:\data\death\deathnames\deathnamesv3.2010" lrecl=241;
   input
      @1   certno      $char10.
      @11  lastname    $char50.
      @61  firstname   $char30.
      @91  middlename  $char40.
      @131 suffix      $char4.
      @158 ssnL4       $char4.
      @162 street      $char35.
      @197 city        $char30.
      @236 statecode   $char2.
      ;
run;
/*
Step 2. merge with standard death file
*/
proc sort data=names;
   by certno;
run;
proc sort data=death.dea2010
      out=stats(keep=certno age dob sex cnty_res zipcode dth_date race_wht race_blk
      race_ami race_asi race_chi race_fil race_gua race_haw race_jap race_kor
      race_opi race_oas race_oth race_sam race_vie hisp zipcode facility fac_type);
   by certno;
run;
/*
combine statistical and name files, and recode race fields to match the reduced
set in the CHARS file (which has only white, black, american indian or alaska
native, asian, hawaiian or other Pacific Islander)
*/
data dwnames(drop=sum_race_asi sum_race_haw race_temp1 race_temp2 race_chi race_fil
          race_gua race_jap race_kor race_opi race_oas race_oth race_sam
          race_vie firsttemp lasttemp middlename hisp firsttemp2 lasttemp2);
   length firstname lastname $ 20 miname hispanic $ 1 lastname_sdx firstname_sdx $ 4
          firsttemp2 lasttemp2 $ 25;
   merge stats(rename=(race_asi=race_temp1 race_haw=race_temp2))
         names(rename=(firstname=firsttemp lastname=lasttemp));
   by certno;
   firsttemp2 = compress(firsttemp," ''-_,.&");
   lasttemp2  = compress(lasttemp," ''-_,.&");
   firstname = substr(firsttemp2,1,20);
   lastname  = substr(lasttemp2,1,20);
   miname    = substr(middlename,1,1);
   sum_race_asi = min(1,(race_chi='Y')+(race_fil='Y')+(race_jap='Y')+(race_kor='Y')+
                  (race_oas='Y')+(race_vie='Y')+(race_temp1='Y'));
   sum_race_haw = min(1,(race_gua='Y')+(race_opi='Y')+(race_sam='Y')+(race_temp2='Y'));

   if sum_race_asi = 0 then race_asi = 'N';
   else                     race_asi = 'Y';
   if sum_race_haw = 0 then race_haw = 'N';
   else                     race_haw = 'Y';
```

```
   if race_ami in ('') then race_ami = 'U';
   if race_asi in ('') then race_asi = 'U';
   if race_blk in ('') then race_blk = 'U';
   if race_haw in ('') then race_haw = 'U';
   if race_wht in ('') then race_wht = 'U';
   if ssnL4 = '9999'      then ssnL4 = '';
   select(hisp);
      when('0')                  hispanic = 'N';
      when('1','2','3','4','5') hispanic = 'Y';
      when('','9')               hispanic = 'U';
      end;
   lastname_sdx  = soundex(lastname);
   firstname_sdx = soundex(firstname);

   format dob mmddyy10.;
run;
```

Listing 14: Create CHARS file for linking

```
data clink0910(keep=seq_no_enc adm_date age country countyres dis_date dob firstname
               ssnL4 hispanic hospital lastname miname race_ami race_asi race_blk
               race_haw race_wht sex statecode status zipcode zipplus4
               lastname_sdx firstname_sdx suffix);
   length firstname lastname $ 20 suffix $ 4 lastname_sdx firstname_sdx $ 4 statecode $ 2;
   set chars.chr_r2009(rename=(SSN=ssnL4 firstname=firsttemp lastname=lasttemp))
       chars.chr_r2010(rename=(SSN=ssnL4 firstname=firsttemp lastname=lasttemp));
   if race_ami in ('','R') then race_ami = 'U';
   if race_asi in ('','R') then race_asi = 'U';
   if race_blk in ('','R') then race_blk = 'U';
   if race_haw in ('','R') then race_haw = 'U';
   if race_wht in ('','R') then race_wht = 'U';
   if hispanic in ('','R') then hispanic = 'U';
   if ssnL4 = '9999'      then ssnL4 = '';
/*
remove the suffixes II, III, IV, V, VI, VII, VIII, ESQ, JR, and SR
from lastnames and place them in a separate suffix field.
Used with UB04 data.
*/
   if _N_ = 1 then do;
    retain __re __reIII;
    pattern = "/( II| III| IV| V| VI| VII| VIII| ESQ|.JR|.SR)$/i";
    __re = prxparse(pattern);
    __reIII = prxparse('/III$/');
   end;
   lasttemp = translate(lasttemp,' ','.,');
   call prxsubstr(__re, TRIM(lasttemp), position, length);
   if position ^= 0 then do;
    suffix    = substr(lasttemp, position + 1, length - 1);
    lasttemp2 = substr(lasttemp, 1, position - 1);
   end;
   else lasttemp2 = lasttemp;

   firstname = compress(firsttemp," ''-_,.&");
   lastname  = compress(lasttemp2," ''-_,.&");
   lastname_sdx  = soundex(lastname);
   firstname_sdx = soundex(firstname);
*   statecode = put(stateres,$stateres.);
   statecode = stateres;
*   if not ('01' le statecode le '69') then statecode = '99';
   if statecode = 'XX' then statecode = '';
run;
```

Listing 15: compute test link scores

```
libname dihd 'c:\data\dihd';

/*
For each record, I will evaluate its similarity with each of the other records
```

```
by computing a score using the points described above. In the output dataset,
I will keep records that have a score of at least 0.
Maximum score is 112.
*/
data dihd.link2010;
   set clink0910(rename=(
     age           = c_age
     countyres     = c_cnty_res
     dob           = c_dob
     firstname     = c_firstname
     hispanic      = c_hispanic
     lastname      = c_lastname
     miname        = c_miname
     race_ami      = c_race_ami
     race_asi      = c_race_asi
     race_blk      = c_race_blk
     race_haw      = c_race_haw
     race_wht      = c_race_wht
     sex           = c_sex
     zipcode       = c_zipcode
     firstname_sdx = c_firstname_sdx
     lastname_sdx  = c_lastname_sdx
     ssnl4         = c_ssnl4
     statecode     = c_statecode
     ));
   do i = 1 to 49190;
      set dwnames point=i;
      score =
      (age           = c_age           and age           ne .)*5  +
      (age           ne c_age          )*(-5) +
      (cnty_res      = c_cnty_res      and cnty_res      ne '')*3  +
      (cnty_res      ne c_cnty_res     )*(-5) +
      (dob           = c_dob           and dob           ne .)*20  +
      (dob           ne c_dob          )*(-20) +
      (month(dob)    = month(c_dob)    and dob           ne .)*3   +
      (month(dob)    ne month(c_dob)   )*(-5)  +
      (day(dob)      = day(c_dob)      and dob           ne .)*4   +
      (day(dob)      ne day(c_dob)     )*(-4)   +
      (year(dob)     = year(c_dob)     and dob           ne .)*4   +
      (year(dob)     ne year(c_dob)    )*(-4)   +
      (firstname     = c_firstname     and firstname     ne '')*10 +
      (firstname     ne c_firstname    )*(-10) +
      (hispanic      = c_hispanic      and hispanic      ne '')*5  +
      (hispanic      ne c_hispanic     )*(-5) +
      (lastname      = c_lastname      and lastname      ne '')*15  +
      (lastname      ne c_lastname     )*(-15) +
      (miname        = c_miname        and miname        ne '')*2  +
      (miname        ne c_miname       )*(-3) +
      (race_ami      = c_race_ami      and race_ami      ne '')*5  +
      (race_ami      ne c_race_ami     )*(-5) +
      (race_asi      = c_race_asi      and race_asi      ne '')*5  +
      (race_asi      ne c_race_asi     )*(-5) +
      (race_blk      = c_race_blk      and race_blk      ne '')*5  +
      (race_blk      ne c_race_blk     )*(-5) +
      (race_haw      = c_race_haw      and race_haw      ne '')*5  +
      (race_haw      ne c_race_haw     )*(-5) +
      (race_wht      = c_race_wht      and race_wht      ne '')*5  +
      (race_wht      ne c_race_wht     )*(-5) +
      (sex           = c_sex           and sex           ne '')*2  +
      (sex           ne c_sex          )*(-20) +
      (zipcode       = c_zipcode       and zipcode       ne '')*3  +
      (zipcode       ne c_zipcode      )*(-2) +
      (firstname_sdx = c_firstname_sdx and firstname_sdx ne '')*2  +
      (firstname_sdx ne c_firstname_sdx)*(-10) +
      (lastname_sdx  = c_lastname_sdx  and lastname_sdx  ne '')*4  +
      (lastname_sdx  ne c_lastname_sdx )*(-10) +
      (ssnl4         = c_ssnl4         and ssnl4         ne '')*15  +
      (ssnl4         ne c_ssnl4        )*(-10) +
```

```
      (statecode     = c_statecode     and statecode       ne '')*1  +
      (statecode     ne c_statecode    )*(-5) +
      (status = '20' and dth_date = dis_date)*10 +
      (status = '20' and dth_date ne dis_date)*(-10) +
      (status = '20' and facility = substr(hospital,1,3))*5 +
      (status = '20' and facility ne substr(hospital,1,3))*(-10) +
      (dis_date ge dth_date + 2)*(-20)
      ;

      if score ge 0 then output;
      *output;

      end;
run;
proc print data=dihd.link2010(obs=21 );
   var score certno firstname c_firstname lastname c_lastname miname c_miname dob c_dob
       ssnL4 c_ssnL4 age c_age sex c_sex cnty_res c_cnty_res statecode c_statecode
       race_ami c_race_ami race_asi c_race_asi race_blk c_race_blk race_haw c_race_haw
       race_wht c_race_wht hispanic c_hispanic;
run;
/*
did any pairs have a high score without birthdate matching?
*/
proc freq data=dihd.link2010;
   where dob ne c_dob;
   tables score;
run;
```

Listing 16: write death and CHARS files to csv for R

```
/*
the length statements are to ensure fields are in a consistent order
when I read them into R, and the fields are ordered for easiest use
during the classification of the training set.
I delete the records that have no names or SSN (typically these are
deaths that occurred out-of-state).
*/
data dwnames2;
   length certno $ 10 dob 8 firstname $ 20 miname $ 1 lastname $ 20
          suffix $ 4 ssnL4 $ 4 sex $ 1 zipcode $ 5 cnty_res $ 2 facility $ 3
          dth_date 8 hispanic race_wht race_blk race_ami race_asi
          race_haw $ 1 statecode $ 2 deathfirst $ 20 deathlast $ 20;
   set dwnames;
   keep certno cnty_res dob firstname hispanic lastname miname race_ami
    race_asi race_blk race_haw race_wht sex ssnL4 statecode
        zipcode facility dth_date deathfirst deathlast suffix
        ;
   if firstname in ('B','BABY','BABYBOY','BABYGIRL','BOY','GIRL')
      then firstname = '';
   deathfirst = firstname;
   deathlast  = lastname;
   if firstname = '' and lastname = '' then delete;
   if hispanic = 'U' then hispanic = '';
   if race_wht = 'U' then race_wht = '';
   if race_blk = 'U' then race_blk = '';
   if race_ami = 'U' then race_ami = '';
   if race_asi = 'U' then race_asi = '';
   if race_haw = 'U' then race_haw = '';
   if sex = 'U' then sex = '';
   if statecode = '99' then statecode = '';
   if zipcode = '99999' then zipcode = '';
   if facility in ('899','999') then facility = '';
   if ssnL4 = '9999' then ssnL4 = '';
   format dth_date mmddyy10.;
run;
proc export data=dwnames2
   outfile = "c:\data\DIHD\death2010.txt"
   dbms = csv
```

```
   replace
   ;
run;

data clink2;
   length seq_no_enc $ 10 dob 8 firstname $ 20 miname $ 1 lastname $ 20
          suffix $ 4 ssnL4 $ 4 sex $ 1 zipcode $ 5 countyres $ 2 facility $ 3
          dth_date 8 hispanic race_wht race_blk race_ami race_asi
          race_haw $ 1 statecode $ 2 charslast $ 20 charsfirst $ 20;
   set clink0910;
   if status = '20' then do;
      facility = substr(hospital,1,3);
      dth_date = dis_date;
      end;
   else do;
      facility = '';
      dth_date = .;
      end;
   if firstname in ('B','BABY','BABYBOY','BABYGIRL',
      'BOY','GIRL','BB','BBABY','BABYA','BABYB','BABYBOY',
      'BABYG','BABYGIRL','BABYTWIN','BABYABOY','BABYBGIRL',
      'BABYBOY','BABYBOYA','BABYBOYB','BABYFEMAL','BABYFEMALE',
      'BABYGIRL','BABYGIRLA','BABYGIRLB','BABYMALE','BABYONE',
      'BABYTWO')
      then firstname = '';
   charsfirst = firstname;
   charslast = lastname;
   if hispanic = 'U' then hispanic = '';
   if race_wht = 'U' then race_wht = '';
   if race_blk = 'U' then race_blk = '';
   if race_ami = 'U' then race_ami = '';
   if race_asi = 'U' then race_asi = '';
   if race_haw = 'U' then race_haw = '';
   if sex = 'U' then sex = '';
   if statecode = '99' then statecode = '';
   if zipcode = '99999' then zipcode = '';
   if facility in ('899','999') then facility = '';
   if ssnL4 = '9999' then ssnL4 = '';
   keep seq_no_enc countyres dob firstname hispanic lastname miname race_ami
    race_asi race_blk race_haw race_wht sex ssnL4 statecode
        zipcode facility dth_date charsfirst charslast suffix;
   format dth_date mmddyy10.;
run;
proc export data=clink2
   outfile = "c:\data\DIHD\chars2009_2010.txt"
   dbms = csv
   replace
   ;
run;

%<<>>=
library(RecordLinkage)

# save the previous training set, model, and results
save(list=c('pairs2011.fsWt','train2011.a','model2011.bag','result2011.bag',
       'manualreview2011.b','predictions.2011b','links2011'),file='Classifier2011')

death2010 <- read.csv("../../../data/DIHD/death2010.txt",colClasses=c(rep("character",18)),
                    col.names=c("certno","dob","firstname","miname","lastname","suffix",
                    "ssnL4","sex","zipcode","county","facility","deathdate","hispanic",
                    "race.wht","race.blk","race.ami","race.asi","race.haw","statecode",
                    "death.first","death.last"))
death2010$firstname.sdx <- soundex(death2010$firstname)
death2010$lastname.sdx  <- soundex(death2010$lastname)

chars0910 <- read.csv("../../../data/DIHD/chars2009_2010.txt",colClasses=c(rep("character",18)),
                    col.names=c("seq_no_enc","dob","firstname","miname","lastname","suffix",
                    "ssnL4","sex","zipcode","county","facility","deathdate","hispanic",
```

```
                    "race.wht","race.blk","race.ami","race.asi","race.haw","statecode",
                    "chars.last","chars.first"))
chars0910$firstname.sdx <- soundex(chars0910$firstname)
chars0910$lastname.sdx  <- soundex(chars0910$lastname)

pairs2010 <- compare.linkage(death2010,chars0910,blockfld=c(2),exclude=c(1))

# calculate Fellegi-Sunter weights
pairs2010.fsWt <- fsWeights(pairs2010)

# use the model that was trained on the 2011 data.
result2010.bag <- classifySupv(model2011.bag,newdata=pairs2010.fsWt)

manualreview2010 <- result2010.bag[(result2010.bag$prediction=='L'&result2010.bag$Wdata<=30)|
                    (result2010.bag$prediction=='N'&result2010.bag$Wdata>=30)]

manualreview2010 <- editMatch(manualreview2010)

%@
```

This shows the relationships between the weight, the machine prediction, and the manual classification on the records that I manually reviewed:

```
> with(manualreview2010[manualreview2010$Wdata<0],table(pairs$is_match,prediction))
   prediction
     N  P  L
  0  0  0  7
  1  0  0 45
> with(manualreview2010[manualreview2010$Wdata>=0&manualreview2010$Wdata<10],table(pairs$is_match,prediction))
   prediction
     N  P  L
  0  0  0 17
  1  0  0 80
> with(manualreview2010[manualreview2010$Wdata>=10&manualreview2010$Wdata<20],table(pairs$is_match,prediction))
   prediction
      N   P   L
  0   0   0  20
  1   0   0 182
> with(manualreview2010[manualreview2010$Wdata>=20&manualreview2010$Wdata<30],table(pairs$is_match,prediction))
   prediction
      N    P    L
  0   0    0    9
  1   0    0  430
> with(manualreview2010[manualreview2010$Wdata>=30&manualreview2010$Wdata<40],table(pairs$is_match,prediction))
   prediction
      N   P  L
  0  75   0  0
  1  15   0  0
> with(manualreview2010[manualreview2010$Wdata>=40&manualreview2010$Wdata<50],table(pairs$is_match,prediction))
   prediction
     N  P  L
  0 28  0  0
  1 31  0  0
> with(manualreview2010[manualreview2010$Wdata>=50&manualreview2010$Wdata<60],table(pairs$is_match,prediction))
   prediction
     N  P  L
  1 19  0  0
> with(manualreview2010[manualreview2010$Wdata>=60],table(pairs$is_match,prediction))
   prediction
    N P L
  1 2 0 0
```

In tabular form:

When the machine predicted a link (I manually reviewed all pairs where the weight was 30 or less and the machine predicted a link):

weight	match	not	% not match
< 0	45	7	13
0–10	80	17	18
10–20	182	20	10
20–30	430	9	2

When the machine predicted a pair was not a link (I manually reviewed all pairs where the weight was 30 or more and the machine predicted the pair was not a link):

weight	match	not	% match
30–40	15	75	17
40–50	31	28	53
50–60	19	0	100
60 +	2	0	100

```
%<<>>=
library(RecordLinkage)

manualreview2010.b <- manualreview2010
for(i in 1:length(manualreview2010$prediction)) {
      manualreview2010.b$prediction[i] <- if(manualreview2010$pairs$is_match[i]==0) 'N' else 'L'
}

predictions.2010a <- result2010.bag$prediction
index.r <- as.numeric(row.names(manualreview2010.b$pairs))
predictions.2010b <- predictions.2010a
predictions.2010b[index.r] <- manualreview2010.b$prediction

# combine death and CHARS row numbers with the predictions
newresults2010 <- cbind(result2010.bag$pairs[,c(1,2)],predictions.2010b)

# get death certificate numbers and CHARS seq number (seq_no_enc)
deathcerts <- result2010.bag$data1[newresults2010[,1],1]
charsseq <- result2010.bag$data2[newresults2010[,2],1]
newresults2010.b <- data.frame(deathcerts,charsseq,predictions.2010b)

%@
```

Manual review of records with non-matching birthdates

Now I get a file of the record pairs which had a high matching score (20 or higher) with non-matching birthdates, and export them to an Excel spreadsheet to conduct a manual review on them. I chose 20 as the cutoff score for manual review because that provides a reasonable number of records for review (about 2,400 for 2010), but I think it includes nearly all the records that have much chance of being classified a true match.

Listing 17: get non-matching birthdate high scorers for manual review

```
libname dihd 'c:\data\dihd';
data review1;
   set dihd.link2010(where=(dob ne c_dob and score ge 20));
run;
proc sort data=review1;
   by score;
run;
data review2(keep=dcert cseq bd fname mi lname ssn sx hosp dd zip county hisp rw
                  rb ram ras rh sc);
   length dcert $ 10 cseq $ 10 bd 8 fname $ 20 mi $ 1 lname $ 20 ssn $ 4 sx $ 1
```

```
        hosp $ 3 dd 8 zip $ 5 county $ 2 hisp rw rb ram ras rh $ 1
        sc 8;
set review1;
format bd dd mmddyy10.;

dcert = certno;
cseq = seq_no_enc;
bd = dob;
fname = firstname;
mi = miname;
lname = lastname;
ssn = ssnL4;
sx = sex;
hosp = facility;
dd = dth_date;
zip = zipcode;
county = cnty_res;
hisp = hispanic;
rw = race_wht;
rb = race_blk;
ram = race_ami;
ras = race_asi;
rh = race_haw;
sc = score;
output;

bd = c_dob;
fname = c_firstname;
mi = c_miname;
lname = c_lastname;
ssn = c_ssnL4;
sx = c_sex;
if status = 20 or dis_date ge dth_date then do;
   hosp = hospital;
   dd = dis_date;
   end;
else do;
   hosp = '';
   dd = .;
   end;
zip = c_zipcode;
county = c_cnty_res;
hisp = c_hispanic;
rw = c_race_wht;
rb = c_race_blk;
ram = c_race_ami;
ras = c_race_asi;
rh = c_race_haw;
sc = .;
output;

bd = .;
fname = '';
mi = '';
lname = '';
ssn = '';
sx = '';
hosp = '';
dd = .;
zip = '';
county = '';
hisp = '';
rw = '';
rb = '';
ram = '';
ras = '';
rh = '';
sc = .;
```

```
   output;
run;

proc export data=review2
   outfile = "c:\user\projects\Death-CHARSlink\manreview2010.xls"
   dbms = excel5
   replace
   ;
run;
/*
read the reviewed links
*/
proc import out=review3
   file = "c:\user\projects\Death-CHARSlink\manreview2010_done.xls"
   dbms = excel5
   ;
run;
```

Now I need to combine the links from three sources: the machine learning results, the manual review of those results, and the manual coding of the records on which birthdate didn't match. After combining those links, I need to check whether there are any hospitalization records linked to more than one death record, and if so, adjudicate those links manually. Then I can create the final linked file.

```
#create file containing only the linked pairs
links2010 <- newresults2010.b[newresults2010.b$predictions.2010b=='L',]

write.csv(links2010,file="c:/data/dihd/links2010.csv",row.names=F)
```

Listing 18: create final linked file for 2010

```
libname dihd 'c:\data\dihd';

proc import out=links0
   file = "c:\data\dihd\links2010.csv"
   dbms = csv
   replace
   ;
run;
data links1(keep=certno seq_no_enc predict);
   length certno seq_no_enc $ 10 predict $ 1;
   set links0;
   certno = substr(deathcerts,1,10);
   seq_no_enc = substr(charsseq,1,10);
   predict = substr(predictions_2010b,1,1);
run;
/*
find the CHARS records that linked to more than one death certificate
(there are 4 CHARS records that each linked to 2 death certs)
*/
proc freq data=links1 noprint;
   tables seq_no_enc/out=charslist;
run;
data mults1(drop=percent);
   set charslist(where=(count ge 2));
run;
proc sort data=links1;
   by seq_no_enc;
run;
data mults2;
   merge links1 mults1(in=inmult);
   by seq_no_enc;
   if inmult;
run;
/*
I'll guess that all these pairs are in the dataset with high scores,
```

```
   and I will get the detailed information from there.
*/
proc sort data=dihd.link2010;
   by certno seq_no_enc;
run;
proc sort data=mults2;
   by certno seq_no_enc;
run;

data mults3;
   merge dihd.link2010 mults2(in=inmult);
   by certno seq_no_enc;
   if inmult;
run;
proc print data=mults3;
run;
/*
I code the pairs by hand and enter the data here
*/
data mults4;
   input @1 certno $char10. @12 seq_no_enc $char10. @23 link $char1.;
datalines;
2010005523 2010406851 N
2010005523 2010603970 L
2010005524 2010406851 L
2010005524 2010603970 N
2010008255 2010059679 N
2010008255 2010611929 L
2010008286 2010059679 L
2010008286 2010611929 N
;;
run;
proc sort data=links1;
   by certno seq_no_enc;
run;
data links2(keep=certno seq_no_enc);
   merge links1 mults4;
   by certno seq_no_enc;
   if link = '' then match = predict;
   else              match = link;
   if match = 'L' then output;
run;

/*
read in the reviewed links for pairs which had high scores but
   non-matching birthdates
*/
proc import out=review3
   file = "c:\user\projects\Death-CHARSlink\manreview2010_done.xls"
   dbms = excel5
   replace
   ;
run;
data mlinks1(keep=certno seq_no_enc sc link);
   length certno seq_no_enc $ 10;
   retain i 0;
   set review3;
   certno = substr(dcert,1,10);
   seq_no_enc = substr(cseq,1,10);
   i+1;
   if i = 1 then output;
   if i = 3 then i = 0;
run;
proc freq data=mlinks1;
   tables sc*link/norow nocol nopercent;
run;
/*
this table shows the strong relation between score and link status
```

The SAS System
11:35 Wednesday, July 16, 2014 2012

The FREQ Procedure

Table of SC by LINK

SC(SC) LINK(LINK)

Frequency	0	1	Total
20	11	6	17
21	63	6	69
22	572	3	575
23	3	5	8
24	330	0	330
25	5	13	18
26	10	2	12
27	100	0	100
28	0	9	9
29	31	0	31
30	32	9	41
31	7	5	12
32	68	2	70
33	0	8	8
34	2	1	3
35	7	15	22
36	0	3	3
37	1	2	3
38	1	29	30
39	6	1	7
40	3	35	38
41	0	1	1
42	1	7	8
43	0	14	14
44	0	7	7
45	1	52	53
46	0	2	2
47	0	5	5
48	0	3	3

```
+--------+--------+--------+
|     49 |      0 |      5 |      5
+--------+--------+--------+
|     50 |      1 |     43 |     44
+--------+--------+--------+
|     51 |      0 |      1 |      1
+--------+--------+--------+
|     52 |      0 |     21 |     21
+--------+--------+--------+
|     53 |      0 |      9 |      9
+--------+--------+--------+
|     55 |      0 |     86 |     86
+--------+--------+--------+
|     56 |      0 |      2 |      2
+--------+--------+--------+
|     57 |      0 |     14 |     14
+--------+--------+--------+
|     58 |      0 |      4 |      4
+--------+--------+--------+
|     60 |      0 |    126 |    126
+--------+--------+--------+
|     61 |      0 |      1 |      1
+--------+--------+--------+
|     62 |      0 |      6 |      6
+--------+--------+--------+
|     63 |      0 |     19 |     19
+--------+--------+--------+
|     64 |      0 |      1 |      1
+--------+--------+--------+
|     65 |      0 |    222 |    222
+--------+--------+--------+
|     67 |      0 |     12 |     12
+--------+--------+--------+
|     68 |      0 |      7 |      7
+--------+--------+--------+
|     70 |      0 |     65 |     65
+--------+--------+--------+
|     71 |      0 |      1 |      1
+--------+--------+--------+
|     72 |      0 |      1 |      1
+--------+--------+--------+
|     73 |      0 |     17 |     17
+--------+--------+--------+
|     75 |      0 |    156 |    156
+--------+--------+--------+
|     77 |      0 |      1 |      1
+--------+--------+--------+
|     78 |      0 |      3 |      3
+--------+--------+--------+
|     80 |      0 |     28 |     28
+--------+--------+--------+
|     82 |      0 |      3 |      3
+--------+--------+--------+
|     85 |      0 |      9 |      9
+--------+--------+--------+
|     88 |      0 |      6 |      6
+--------+--------+--------+
|     90 |      0 |     35 |     35
+--------+--------+--------+
|  Total |   1255 |   1149 |   2404
*/
data mlinks2(keep=certno seq_no_enc);
   set mlinks1(where=(link=1));
run;
data dihd.finallink2010;
   set links2 mlinks2;
run;
```

```
proc freq data=dihd.finallink2010 noprint;
   tables certno/out=dcertlist;
run;
/*
I found that 34,571 death certificates linked to 87,536
hospital records.
*/
```

year 2009

The 2008 CHARS data is not complete enough to link with the death data, so we cannot create a DIHD file for the 2009 deaths.

■ *end analysis details*