Combining genetic and non-genetic data in identification cases

Thore Egeland

Updated: 2022-05-13

Motivation

The DNA based solutions may by symmetric. For instance, if two brothers are missing, it may be impossible to determine using DNA which is the missing person. Age information may resolve such symmetries. Furthermore, information on age (or other non-DNA information) may contain valuable information that can be used beyond ranking otherwise symmetric solutions.

Continuous data

Assume we have non-DNA data $x = (x_1, \ldots, x_s)$ from the victims and similar data $y = (y_1, \ldots, y_m)$ from the missing persons. In the example below x and y record age information. Then x would typically be uncertain estimates whereas the y in most cases (as below) can be considered to be certain.

Ranking

The goal is to rank assignments with the same number of victims, n, identified. Obviously, $n \leq \max\{s, m\}$. Consider an assignment a that specifies the identification of n victims and let $d_1(a), \ldots, d_n(a)$ be the differences between the ages of victims and the missing persons. Let

$$Q(a) = \sum_{i=1}^{d} d_i(a)^2 \tag{1}$$

It is intuitively reasonable to prefer the solution with the smallest $Q(a)$ since then age information indicates the closest match. The minimum value $Q(a) = 0$ is obtained when the ages of the missing persons and victims match perfectly. Ranking can be done based on $Q(a)$. Obviously weighted versions like

$$Q_w(a) = \sum_{i=1}^{d} w_i d_i^2 \tag{2}$$

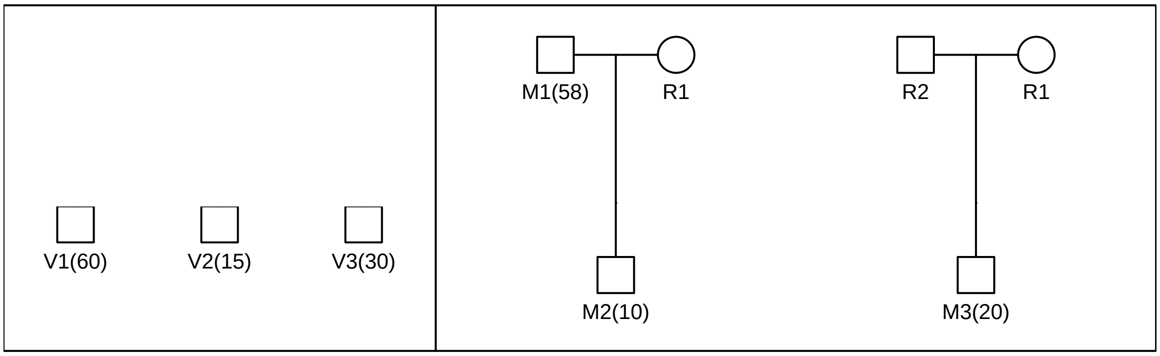
can be used to reflect that uncertainty may depend on age.

Example 1

There are three victims, V_1, V_2, V_3, with ages estimated to 60, 15 and 30, and three missing persons M_1, M_2, M_3, with ages 58, 10 and 20. All victims and missing persons are males, as shown in the below figure.

```
library(dvir, quietly = T)
library(dvicomb, quietly =T)
# source("C://Users//theg//Dropbox//umb//master//alfErik//alfR//dvirCode//AgeAIC.R")
# source("C://Users//theg//Dropbox//umb//master//alfErik//alfR//dvirCode//expand.grid.nodup2.R")
# source("C://Users//theg//Dropbox//umb//master//alfErik//alfR//dvirCode//LRdiscrete.R")
```

```
pm = example2$pm
am = example2$am
missing = example2$missing
am[1] = setAlleles(am[1],"R1", alleles = 0)
pm[[3]] = swapSex(pm[[3]], "V3")
am[[2]] = swapSex(am[[2]], "M3")
pm = setMutationModel(pm, model = "proportional", rate = 0.01)
am = setMutationModel(am, model = "proportional", rate = 0.01)
jointRes = jointDVI(pm, am, missing, disableMutations = FALSE, verbose = F)
pmP = pm
pmP[[1]]$ID = "V1(60)"
pmP[[2]]$ID = "V2(15)"
pmP[[3]]$ID = "V3(30)"
amP = am
amP[[1]] = relabel(amP[[1]], c("M1(58)", "R1", "M2(10)"))
amP[[2]] = relabel(amP[[2]], c("R2", "R1", "M3(20)"))
plotPedList(list(pmP, amP))
```



Consider the below table. The table gives the assignments with one ($n = 1$) victim identified. For instance, the best one is $a_2 = (V_1 = M_1, V2 = *, V_3 = *)$ with $Q(a_2) = (58 - 60)^2 = 4$.

```
n = apply(jointRes[,1:3],1 , function(x) 3-length(x[x=='*']))
jointRes = data.frame(jointRes,n)
jointRes = jointRes[order(jointRes$n),]
n = jointRes$n
```

Assume the following user input

```
aM = list(M1 = 58, M2 = 10, M3 = 20)
aV = list(V1 = 60, V2 = 15, V3 = 30)
sigma = 2
```

```
x = matrix(ncol =3, nrow = 34)
Q = loglikAge = rep(NA, 34)
for (i in 1:34){
  for (j in 1:3)
    x[i,j] = aV[[j]]-ifelse(jointRes[i,j] =='*', NA, as.double(aM[jointRes[i,j]]))
  Q[i] = sum(x[i,]^2, na.rm = T)
  if(all(is.na(x[i,]))) Q[i] = NA
  loglikAge[i] = -n[i]/2*log(2*pi)-n[i]*log(sigma)-0.5*Q[i]/sigma^2
}
```

```
jointRes = data.frame(jointRes, Q, loglikAge)
resOrdered = jointRes[order(jointRes$n, jointRes$Q),]
resOrdered2 = resOrdered[,c(1:3, 7, 8,9)]
rownames(resOrdered2) = NULL
dat = resOrdered2
dat[2:10, 1:5]
```

```
##    V1 V2 V3 n    Q
## 2  M1  *  * 1    4
## 3   * M2  * 1   25
## 4   * M3  * 1   25
## 5   *  * M3 1  100
## 6   *  * M2 1  400
## 7   *  * M1 1  784
## 8  M3  *  * 1 1600
## 9   * M1  * 1 1849
## 10 M2  *  * 1 2500
```

Beyond ranking: Probabilistic approach

Consider an assignment a with $\mid a \mid \leq \max\{s, m\}$ victims identified. Let

$$\{(i_1, j_1), \ldots, (i_t, j_t)\}$$

describe the pairwise identifications, i.e.,:

$$V_{i_1} = M_{j_1}, \ldots V_{i_t} = M_{j_t}$$

We assume that the age of an identified victim is normally distributed with standard deviation σ and expectation equal to the corresponding missing person. For an an unidentified victim, all ages are equally likely and we assume a uniform distribution on $[0, k]$, default $k = 100$. We assume conditional independence given the assignment. The likelihood can be written

$$\begin{aligned} L(a) &= L(x_1, \ldots, x_s \mid a, y_1, \ldots, y_m) \\ &= \left(\frac{1}{k}\right)^{s-|a|} L(x_{i_1} \mid y_{j_1}) \cdots L(x_{i_t} \mid y_{j_t}) \end{aligned}$$

The log likelihood is

$$l(a) = -(s- \mid a \mid) \log(k) + \log(\phi(x_{i_1}, y_{j_1}, \sigma^2) + \cdots + \log(\phi(x_{i_t}, y_{j_t}, \sigma^2) \tag{3}$$

where ϕ is the pdf of the normal distribution.

The Akaike Information Criterion (AIC) is an alternative measure to compare models. The idea is to punish models with many parameters estimated to avoid overfitting, i.e., spurious identification. It is not obvious how AIC should be defined for the present application. We have tentatively defined

$$AIC(a) = 2 \mid a \mid -2l(a) \tag{4}$$

The assignment with the smallest AIC is preferred. We also report the likelihood ratio of an assignment using the null assignment (no victims identified) as reference.

A Bayesian approach requires a prior for each assignment. Below we exemplify using a flat prior. In this way we can see the impact of age information on the posterior. The posterior could be used as a prior for the DNA based solution.

Likelihood ratio

We can alternatively consider the likelihood ratio (LR) comparing an assignment a to the null assignment(no victims identified):

$$
\begin{aligned}
LR_a &= LRi_1 \cdots LR_{i_t} \\
&= \frac{\phi(x_{i_1}, y_{j_1}, \sigma^2)}{\frac{1}{k}} \cdots \frac{\phi(x_{i_t}, y_{j_t}, \sigma^2)}{\frac{1}{k}} \\
&= \left(k \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} e^{-\frac{1}{2}\left(\frac{x_{i_1} - y_{i_1}}{\sigma}\right)^2} \right) \cdots \left(k \frac{1}{\sqrt{2\pi}} \frac{1}{\sigma} e^{-\frac{1}{2}\left(\frac{x_{i_t} - y_{i_t}}{\sigma}\right)^2} \right) \qquad (5)
\end{aligned}
$$

We see that $LR_i > 1$ if

$$k > \sqrt{2\pi}\sigma e^{\frac{1}{2}\left(\frac{x_i - y_i}{\sigma}\right)^2}.$$

Make a figure to illustrate

Example 2

The example below continues the previous. The ages of the missing persons are as in the above figure. The missing person ages are simulated according to the model described above giving (the code is given in the documentation of `dvir::ageAIC`).

```r
set.seed(177)
miss = c('*', missing)
lst = list(V1 = miss, V2 = miss, V3 = miss)
tab = expand.grid.nodup2(lst, pm, am)
ageM = list(M1 = 58, M2 = 10, M3 = 20)

# Simulate ages from assignment (V1 = M1, V2 = M2)
set.seed(177)
sigma = 2
ageV = list(V1 = rnorm(1, 58, sigma), V2 = rnorm(1, 10, sigma),
         V3 = runif(1, 0, 100))
output = round(as.double(ageV),1)
names(output) = missing
output
```

```
##   M1   M2   M3
## 56.3 12.0 44.3
```

The assignment from which victim ages were simulated comes out as the most likely according to all criteria (likelihood, posterior, AIC and LR):

```r
res = ageAIC(tab, ageV, ageM, k = 100, sigma = 2)
# The assignment from which victim ages were simulated
# comes out as the most likely. Below the five best are sorted according to AIC:
log0 = res[1,]$logLikAge
LR = exp(res$logLikAge-log0)
resAge = data.frame(res, LR = LR)
res = resAge[order(resAge$AIC, decreasing = F)[1:5],]
rownames(res) = NULL
cbind(res[,1:3], round(res[,4:8],4))
```

```
##   V1 V2 V3  prior logLikAge posterior     AIC       LR
## 1 M1 M2  * 0.0294   -8.7046    0.8602 21.4091 165.8289
## 2 M1  *  * 0.0294  -11.1806    0.0723 24.3612  13.9421
## 3  * M2  * 0.0294  -11.3395    0.0617 24.6789  11.8941
## 4  *  *  * 0.0294  -13.8155    0.0052 27.6310   1.0000
## 5 M1 M3  * 0.0294  -16.1201    0.0005 36.2401   0.0998
```

LR for discrete data

We consider the hypotheses $H_1:\ V_i = M_i$ and $H_2:\ V_i$ and M_i are unrelated.

Assume $y_i = 1$ if missing person i has a specific identifying property, like a tattoo, and assume $P(Y_i = 1) = 1 - P(Y_i = 0) = \alpha$. Similarly, $P(X_i = 1) = 1 - P(X_i = 0) = \alpha$.

Furthermore, assume $P(X_i = 0 \mid Y_i = 0, H_1) = P(X_i = 1 \mid Y_i = 1, H_1) = 1 - \mu$. In other words, the property is observed in the assigned victim with probability $1-\mu$. If there is discordance in the property of the missing person and the identified victim, and $\mu = 0$, an exclusion results. Furthermore,

$$P(X = 1 \mid Y = 0, H_1) = P(X = 0 \mid Y = 1, H_1) = \mu$$

Let LR_{xy}, x, $y \in \{0,1\}$ denote the likelihood ratio when an identification of a victim is compared to a non-identification. We find

$$\begin{aligned}
LR_{00} &= \frac{P(X_i = 0 \mid Y_i = 0, H_1)}{P(X_i = 0 \mid Y_i = 0, H_2)} = \frac{P(X_i = 0 \mid Y_i = 0, H_1)}{P(X_i = 0)} = \frac{1-\mu}{1-\alpha},\\
LR_{01} &= \frac{P(X_i = 0 \mid Y_i = 1, H_1)}{P(X_i = 0)} = \frac{\mu}{1-\alpha},\\
LR_{10} &= \frac{P(X_i = 1 \mid Y_i = 0, H_1)}{P(X_i = 1)} = \frac{\mu}{\alpha},\\
LR_{11} &= \frac{P(X_i = 1 \mid Y_i = 1, H_1)}{P(X_i = 1)} = \frac{1-\mu}{\alpha},
\end{aligned}$$

With reasonable values for the parameters α and μ we should have

$$LR_{00} > 1, LR_{11} > 1, \Rightarrow \mu < \alpha \text{ and } \mu < 1 - \alpha$$

This implies the similarly reasonable requirement that

$$LR_{01} < 1 \text{ and } LR_{10} < 1.$$

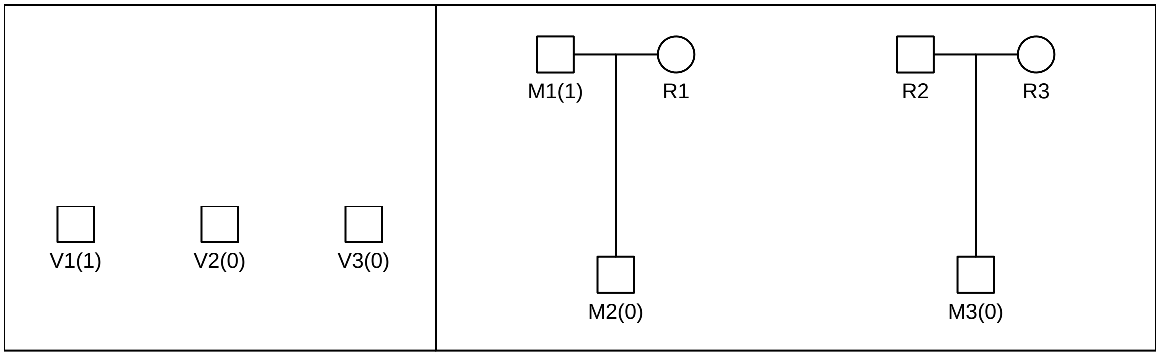
Let $n_{00}, n_{10}, n_{01}, n_{11}$ record the corresponding counts and so for instance n_{00} counts the number of times the property is missing in both the missing person and the victim. The likelihood ratio becomes

$$LR_a = LR_{00}^{n_{00}} LR_{01}^{n_{01}} LR_{10}^{n_{10}} LR_{11}^{n_{11}}. \tag{6}$$

Example 3

We continue the previous example, but now we consider a discrete property as illustrated below. Victim V_1 and the missing person M_1 share a rare property, $P(Y = 1) = \alpha = 0.05$. We set $\mu = 0.01$, meaning that a property in a missing person is observed in the corresponding victim with probability $1 - \mu = 0.99$.

```
pmP[[1]]$ID = "V1(1)"
pmP[[2]]$ID = "V2(0)"
pmP[[3]]$ID = "V3(0)"
amP = am
amP[[1]] = relabel(amP[[1]], c("M1(1)", "R1", "M2(0)"))
amP[[2]] = relabel(amP[[2]], c("R2", "R3", "M3(0)"))
plotPedList(list(pmP, amP))
```



Below the assignments are sorted according to the LR. We have also assumed a prior π, flat in the example, and caclulated the posterior from

$$P(a \mid data) = \frac{\pi_i LR_i}{\sum_i \pi_i LR_i}$$

We see that assignments with $V_1 = M_1$ are sorted on top.

```
x = list(V1 = 1, V2 = 0, V3 = 0)
y = list(M1 = 1, M2 = 0, M3 = 0)
na = dim(tab)[1]
LR = rep(NA, na)
prior = rep(1/na, na)
for (i in 1:na)
 LR[i] = LRDiscrete(tab[i,], x, y, alpha = 0.05, mu = 0.01)
posterior = prior*LR/sum(prior*LR)
resDiscrete = data.frame(tab, prior = round(prior,4), LR = round(LR, 4),
                         posterior = round(posterior,4))
res = resDiscrete[order(resDiscrete$LR, decreasing = T),]
res
```

```
##    V1 V2 V3  prior      LR posterior
## 18 M1 M2 M3 0.0294 21.5025    0.1396
## 20 M1 M3 M2 0.0294 21.5025    0.1396
## 15 M1  * M2 0.0294 20.6337    0.1340
## 16 M1  * M3 0.0294 20.6337    0.1340
## 17 M1 M2  * 0.0294 20.6337    0.1340
## 19 M1 M3  * 0.0294 20.6337    0.1340
## 14 M1  *  * 0.0294 19.8000    0.1286
## 10  * M2 M3 0.0294  1.0860    0.0071
## 13  * M3 M2 0.0294  1.0860    0.0071
## 3   *  * M2 0.0294  1.0421    0.0068
## 4   *  * M3 0.0294  1.0421    0.0068
## 8   * M2  * 0.0294  1.0421    0.0068
## 11  * M3  * 0.0294  1.0421    0.0068
## 1   *  *  * 0.0294  1.0000    0.0065
## 23 M2  * M3 0.0294  0.2084    0.0014
## 26 M2 M3  * 0.0294  0.2084    0.0014
## 30 M3  * M2 0.0294  0.2084    0.0014
## 33 M3 M2  * 0.0294  0.2084    0.0014
## 21 M2  *  * 0.0294  0.2000    0.0013
## 28 M3  *  * 0.0294  0.2000    0.0013
## 6   * M1 M2 0.0294  0.0110    0.0001
## 7   * M1 M3 0.0294  0.0110    0.0001
## 9    * M2 M1 0.0294  0.0110    0.0001
## 12   * M3 M1 0.0294  0.0110    0.0001
## 2    *  * M1 0.0294  0.0105    0.0001
## 5    * M1  * 0.0294  0.0105    0.0001
## 25 M2 M1 M3 0.0294  0.0022    0.0000
## 27 M2 M3 M1 0.0294  0.0022    0.0000
## 32 M3 M1 M2 0.0294  0.0022    0.0000
## 34 M3 M2 M1 0.0294  0.0022    0.0000
## 22 M2  * M1 0.0294  0.0021    0.0000
## 24 M2 M1  * 0.0294  0.0021    0.0000
## 29 M3  * M1 0.0294  0.0021    0.0000
## 31 M3 M1  * 0.0294  0.0021    0.0000
```

We can merge the above table with the previous based on age to get combined LR's and posteriors: (Correct?)

```
LR.age = resAge$LR
LR.discrete = resDiscrete$LR
LR.tot = LR.age * LR.discrete
posterior = prior*LR.tot/sum(prior*LR.tot)
tab2 = data.frame(tab, prior = round(prior,4), LR.tot = LR.tot,
                  posterior = round(posterior,8))
tab2
```

```
##    V1 V2 V3  prior        LR.tot  posterior
## 1    *  *  * 0.0294  1.000000e+00 0.00026931
## 2    *  * M1 0.0294  1.184620e-11 0.00000000
## 3    *  * M2 0.0294  3.942925e-63 0.00000000
## 4    *  * M3 0.0294  2.319120e-31 0.00000000
## 5    * M1  * 0.0294 4.158184e-116 0.00000000
## 6    * M1 M2 0.0294 1.648224e-178 0.00000000
## 7    * M1 M3 0.0294 9.694399e-147 0.00000000
## 8    * M2  * 0.0294  1.239486e+01 0.00333807
## 9    * M2 M1 0.0294  1.476096e-10 0.00000000
## 10   * M2 M3 0.0294  2.874590e-30 0.00000000
## 11   * M3  * 0.0294  7.459777e-03 0.00000201
## 12   * M3 M1 0.0294  8.883799e-14 0.00000000
## 13   * M3 M2 0.0294  2.941409e-65 0.00000000
## 14 M1  *  * 0.0294  2.760535e+02 0.07434423
## 15 M1  * M2 0.0294  1.088464e-60 0.00000000
## 16 M1  * M3 0.0294  6.402048e-29 0.00000000
## 17 M1 M2  * 0.0294  3.421664e+03 0.92149178
## 18 M1 M2 M3 0.0294  7.935294e-28 0.00000000
## 19 M1 M3  * 0.0294  2.059309e+00 0.00055459
## 20 M1 M3 M2 0.0294  8.119748e-63 0.00000000
## 21 M2  *  * 0.0294 1.549988e-116 0.00000000
## 22 M2  * M1 0.0294 1.836147e-127 0.00000000
## 23 M2  * M3 0.0294 3.594264e-147 0.00000000
## 24 M2 M1  * 0.0294 6.445138e-232 0.00000000
## 25 M2 M1 M3 0.0294 1.502621e-262 0.00000000
## 26 M2 M3  * 0.0294 1.156146e-118 0.00000000
## 27 M2 M3 M1 0.0294 1.376979e-129 0.00000000
## 28 M3  *  * 0.0294  1.094244e-71 0.00000000
## 29 M3  * M1 0.0294  1.296263e-82 0.00000000
## 30 M3  * M2 0.0294 4.314108e-134 0.00000000
## 31 M3 M1  * 0.0294 4.550068e-187 0.00000000
## 32 M3 M1 M2 0.0294 1.803559e-249 0.00000000
## 33 M3 M2  * 0.0294  1.356170e-70 0.00000000
## 34 M3 M2 M1 0.0294  1.615209e-81 0.00000000
```