Analysis of longitudinal data with irregular observation times

Eleanor M. Pullenayegum

2016-03-08

Overview

Longitudinal data often has follow-up times that are irregular and potentially related to outcomes. For example, in a clinic-based cohort study where all follow-up is part of usual care, patients may visit more often when unwell. This risks over-estimating the burden of disease unless data are analysed appropriately.

There are two categories of methods available for analysing this type of data: methods based on inverse-intensity weighting, and methods based on semi-parametric joint models (see Pullenayegum and Lim (In Press) for an overview). This package provides methods for inverse-intensity weighting.

Inverse-intensity weighting weights data by the reciprocal of the intensity (or, equivalently, the hazard) of the visit process. Inverse-intensity weighting works in a similar way to survey weighting: observations with a higher intensity are over-represented in the data, and hence should receive less weight. Conversely, observations with lower intensity are under-represented in the data, and hence receive more weight. Lin, Scharfstein, and Rosenheck (2004) show that inverse-intensity weighting followed by a marginal analysis with a generalized estimating equation (GEE) results in unbiased estimation, subject to the assumptions set out below. This package contains functions to compute inverse-intensity weights, and also to fit inverse-intensity weighted GEEs.

Sometimes inference may be desired for a model for which weighting is not straightforward, for example a generalized linear mixed model or a latent class mixed model. In these cases, multiple outputation is a useful alternative to invese-intensity weighting.

Multple outputation works by discarding (outputting) excess data (Hoffman, Sen, and Weinberg 2001; Follmann, Proschan, and Leifer 2003). Visits are randomly deleted from the dataset with probability inversely proportional to the visit intensity (Pullenayegum 2016). The resulting thinned visit process is independent of the outcome process (subject to the assumptions below), and can hence be analysed using standard methods. To avoid wasting data, the random deletion is repeated multiple times and the results from each analysis combined. Conceptually, multiple outputation is the opposite of multiple imputation: where multiple imputation imputes missing observations multiple times, multiple outputation discards excess observations multiple times. This package contains functions to create outputted datasets, as well as to combine results across multiple outputations.

Assumptions

Inverse-intensity methods, whether implemented through weighting or multiple outputation, are subject to the following assumptions.

Suppose $Y_i(t)$ is the outcome of interest for subject i at time t, let $X_i(t)$ be a possibly time-dependent covariate, and suppose interest is in the marginal model

$$E(Y_i(t)|X_i(t)) = X_i(t)\beta.$$

Moreover, suppose that we observe Y_i only at times $T_{i1}, T_{i2}, \ldots, T_{in_i}$. Let $N_i(t)$ be the counting process for the visit times, that is $N_i(t) = \sum_j I(T_{ij} \leq t)$, where $I()$ is an indicator function. Inverse-intensity weighting methods hinge on there being a vector of observed covariates $Z_i(t)$ such that

$$\lim_{\delta \downarrow 0} \frac{E(N(t) - N(t-\delta))|Y_i(t), Z_i(t))}{\delta} = \frac{E(N(t) - N(t-\delta))|Z_i(t))}{\delta} = \lambda(t; Z_i(t))$$

for some hazard function λ. If this assumption is not met, altnerative methods of analysis should be considered (e.g. semi-parametric joint models).

Inverse-Intensity weighting

The first step in inverse-intensity weighting is to calculate the weights, usually by fitting a proportional hazards model to the recurrent event process formed by the visit times. That is, one fits the model

$$\lambda(t; Z_i(t)) = \lambda_0(t)\exp(Z_i(t)\gamma).$$

The inverse-intensity weight is then $w_i(t) = \frac{\exp(-Z_i(t)\gamma)}{\lambda_0(t)}$. Following Buzkova and Lumley (2007), all inverse-intensity weights in this package are stabilized by the baseline hazard $\lambda_0(t)$, so that the stabilized weight is $sw_i(t) = \exp(-Z_i(t)\gamma)$. In some settings one may additionally wish to stabilize by a function of the time-

invariant covariates in $X_i(t)$, and an option to do so is included in the functions. One can then fit a GEE to obtain the regression coefficients β corresponding to regressing $Y_i(t)$ on $X_i(t)$, weighting by $sw_i(t)$.

Example

This package includes data from a randomized trial of thiotepa vs. placebo for the treatment of superficial bladder tumours (Byar and Blackard 1977, Andrews and Herzberg (1985)). At each visit, the number of new tumours (`value`) and the size of the largest tumour (`size`) were recorded. For the purposes of illustration, we focus on the relationship between the size of the largest tumour and randomized group (`Group`), adjusting for time since enrolling in the trial.

```
library(IrregLong)
```

```
data(datalong)
head(datalong)
```

```
##   id basesize Group time value basenum size event
## 1  1        3     0    0     1       1    3     1
## 2  1        3     0    1     0       1    0     1
## 3  2        1     0    0     2       2    1     1
## 4  2        1     0    1     0       2    0     1
## 5  2        1     0    4     0       2    0     1
## 6  3        1     0    0     1       1    1     1
```

```
iiwgee <- iiwgee(formulagee=size~time + Group,
 formulaph=Surv(time.lag,time,event)~ Group*as.numeric(value.lag>1) + cluster(id),
 formulanull=NULL,data=datalong,id="id",time="time",event="event",lagvars=c("time","value"),
 invariant=c("id","basesize","Group","basenum"),maxfu=NULL,first=1)
```

This fits an inverse-intensity weighted GEE, stabilized by just the baseline hazard. In this case the recurrent events model for the visits is a proportional hazards model in which visit intensity is regressed onto randomized group, whether there was more than one new tumour at the last visit, and their interaction. Notice that the recurrent events model (`formulaph`) includes variables that are not in the original dataset. By including time and value in the `lagvars` parameter, the `iiwgee` function creates two new variables, `time.lag` and `value.lag`, that are equal to time and value at the previous visit.

The function returns the inverse-intensity weighted GEE fit:

```
summary(iiwgee$geefit)
```

```
##
## Call:
## geeglm(formula = formulagee, family = family, data = data, weights = useweight,
##     id = id, corstr = "independence")
##
##  Coefficients:
##              Estimate   Std.err    Wald Pr(>|W|)
## (Intercept)  0.761145  0.067807 126.006  < 2e-16 ***
## time        -0.028008  0.003649  58.921 1.64e-14 ***
## Group       -0.127790  0.060661   4.438   0.0352 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Estimated Scale Parameters:
##             Estimate Std.err
## (Intercept)   0.7261  0.1339
##
## Correlation: Structure = independenceNumber of clusters:   85   Maximum cluster size: 31
```

from which we see that the size of the largest tumour was significantly smaller among patients in the thiotepa group as compared to placebo. One can also see the fitted visit intensity model:

```
summary(iiwgee$phfit)
```

```
## Call:
## coxph(formula = formulaph, data = datacox)
##
##   n= 752, number of events= 752
##    (85 observations deleted due to missingness)
```

```
##
##                                 coef exp(coef) se(coef) robust se     z
## Group                         0.8715    2.3904   0.0809    0.1574  5.54
## as.numeric(value.lag > 1)     0.0269    1.0273   0.1396    0.0903  0.30
## Group:as.numeric(value.lag > 1) -0.9490    0.3871   0.2254    0.3072 -3.09
##                                 Pr(>|z|)
## Group                          3.1e-08 ***
## as.numeric(value.lag > 1)         0.766
## Group:as.numeric(value.lag > 1)    0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                                 exp(coef) exp(-coef) lower .95 upper .95
## Group                               2.390      0.418     1.756     3.254
## as.numeric(value.lag > 1)           1.027      0.973     0.861     1.226
## Group:as.numeric(value.lag > 1)     0.387      2.583     0.212     0.707
##
## Concordance= 0.638  (se = 0.016 )
## Rsquare= 0.167   (max possible= 1 )
## Likelihood ratio test= 138  on 3 df,   p=0
## Wald test            = 32.9  on 3 df,   p=3.35e-07
## Score (logrank) test = 148  on 3 df,   p=0,   Robust = 13.9  p=0.00304
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

In practice, it is usually helpful to focus first on choosing the visit intensity model, which can be done by fitting proportional hazards models in the usual way. This may require lagged versions of some of the variables in the dataset, which can be created through the function `lagfn`:

```
data(datalong)
datalong <- lagfn(datalong,c("time","value"),"id","time")
mph <- coxph(Surv(time.lag,time,event)~ Group*as.numeric(value.lag>1) + cluster(id),
 data=datalong[datalong$time>0,])
summary(mph)
```

One can then compute the inverse intensity weights:

```
datalong$weight <- iiw(mph,datalong,"id","time",TRUE)
head(datalong)
```

```
##   id basesize Group time value basenum size event time.lag value.lag
## 1  1        3     0    0     1       1    3     1       NA        NA
## 2  1        3     0    1     0       1    0     1        0         1
## 3  2        1     0    0     2       2    1     1       NA        NA
## 4  2        1     0    1     0       2    0     1        0         2
## 5  2        1     0    4     0       2    0     1        1         0
## 6  3        1     0    0     1       1    1     1       NA        NA
##   weight
## 1  1.000
## 2  1.584
## 3  1.000
## 4  1.542
## 5  1.584
## 6  1.000
```

Note that `time.lag` and `value.lag` were added as columns to `datalong` when `lagfn` was called. If you wish to preserve the data in its original format, you can use `iiw.weights` to fit the visit intensity model and compute the weights:

```
data(datalong)
head(datalong)
```

```
##   id basesize Group time value basenum size event
## 1  1        3     0    0     1       1    3     1
## 2  1        3     0    1     0       1    0     1
## 3  2        1     0    0     2       2    1     1
## 4  2        1     0    1     0       2    0     1
## 5  2        1     0    4     0       2    0     1
## 6  3        1     0    0     1       1    1     1
```

```
weights <- iiw.weights(Surv(time.lag,time,event)~ Group*as.numeric(value.lag>1) + cluster(id),
 formulanull=NULL,data=datalong,id="id",time="time",event="event",lagvars=c("time","value"),
 invariant=c("basesize","Group","basenum"),maxfu=NULL,first=TRUE,frailty=FALSE)
# summary(weights$m)
datalong$weight <- weights$iiw.weight
head(datalong)
```

```
##   id basesize Group time value basenum size event weight
## 1  1        3     0    0     1       1    3     1  1.000
## 2  1        3     0    1     0       1    0     1  1.584
## 3  2        1     0    0     2       2    1     1  1.000
## 4  2        1     0    1     0       2    0     1  1.542
## 5  2        1     0    4     0       2    0     1  1.584
## 6  3        1     0    0     1       1    1     1  1.000
```

Multiple Outputation

Once the inverse-intensity weights have been calculated, an alternative use for them is to perform multiple outputation. The code below computes 5 outptted datasets, analyses each using an unweighted GEE, then combines the results.

```
data(datalong)
weights <- iiw.weights(Surv(time.lag,time,event)~ Group*as.numeric(value.lag>1) + cluster(id),
 Surv(time.lag,time,event)~ Group,data=datalong,id="id",time="time",event="event",
 lagvars=c("time","value"),invariant=c("basesize","Group","basenum"),
 maxfu=NULL,first=TRUE,frailty=FALSE)
reg <- function(data){
    return(data.matrix(summary(geeglm(size~time + Group, id=id,data=data))$coefficients[,1:2]))
}
mo(5,reg,datalong,weights$iiw.weight,singleobs=FALSE,id="id",time="time",keep.first=TRUE)
```

```
## , , 1
##
##       [,1]     [,2]    [,3]
## [1,] 1.263 -0.04910 -0.3094
## [2,] 1.193 -0.04707 -0.2567
## [3,] 1.166 -0.04705 -0.2316
## [4,] 1.183 -0.04490 -0.3106
## [5,] 1.179 -0.04581 -0.2677
##
## , , 2
##
##         [,1]     [,2]    [,3]
## [1,] 0.11566 0.005748 0.09629
## [2,] 0.10880 0.004996 0.09102
## [3,] 0.10298 0.005074 0.08318
## [4,] 0.09837 0.004608 0.09083
## [5,] 0.11061 0.005491 0.09668
```

```
## $est
## [1]  1.19659 -0.04679 -0.27520
##
## $se
## [1] 0.32575 0.07198 0.30109
##
## $RE.MO
## [1] 1.003 1.000 1.003
```

The first set of outputs are the GEE regression coefficients for the intercept, time, and Group for each of the 5 outputted datasets, together with their standard errors. The multiple outputation estimate of the regression coefficients is given by `$est`, and is the mean across outputations of the GEE estimates. The multiple outputation standard error (`$se`) is a function of both the within outputation standard errors and the between outputation variance. The relative efficiency of using 5 outputations in place of all possible outputations is given by `RE.MO`.

Multiple outputation is helpful for analyses where weighting is difficult to implement. One such example is a semi-parametric joint model. Semi-parametric joint models are useful when there are latent variables that influence both the outcome and visit processes. We consider the Liang (Liang, Lu, and Ying 2009) semi-parametric joint model:

$$Y_i(t) = \beta_0(t) + X_i(t)\beta + W_i(t)\nu_{i1} + \epsilon_i(t)$$

$$\lambda_i(t) = \nu_{i2}\lambda_0(t)\exp(U_i\gamma)$$

where $W_i(t)$ is a subset of the covariates $X_i(t)$, U_i is a vector of baseline covariates, $\epsilon_i(t)$ is a mean-zero random error, and ν_{i1}, ν_{i2} are (potentially correlated) random effects.

The Liang model requires that the covariates in the model for the visits be time invariant. In practice,

$$\lambda_i(t) = \nu_{i2}\lambda_0(t)\exp(Z_i(t)\gamma),$$

for an observed vector of time-dependent auxiliary covariates $Z_i(t)$ may be more reasonable. Multiple outputation makes inference under this model possible by creating outputted datasets in which the visit process does not depend on the observed covariates $Z_i(t)$. These datasets can be analysed using Liang's method for the special case where there are no covariates in the visit process model.

Example

We consider the same regression model as before, regressing size onto group and adjusting for time, but this time with a random intercept and a random effect for time. The marginal visit process is the same as before, but we allow the visit and outcome processes to be correlated through random effects as well as through the auxiliary covariates. That is, we take

$$size_i(t) = \beta_0(t) + \beta_1 Group_i + \nu_{i11} + \nu_{i12}t + \epsilon_i(t),$$

with visit process model given by

$$\lambda_i(t) = \nu_{i2}\lambda_0(t)\exp(I(value.lag_i > 1)\gamma_1 + Group_i * I(value.lag_i > 1)\gamma_2),$$

where $\nu_{i1} = (\nu_{i11}, \nu_{i12})'$ and ν_{i2} are potentially correlated. Note that when calculating the weights used for outputation, the `frailty=TRUE` option should be used.

```
Liangmo <- function(data,Yname,Xnames,Wnames,maxfu,baseline){
 x <- Liang(data,Yname,Xnames,Wnames,id="id",time="time",maxfu,baseline); print(x); return(x)
}
weights <-
 iiw.weights(Surv(time.lag,time,event)~ Group*as.numeric(value.lag>1) + cluster(id),
   Surv(time.lag,time,event)~ Group + cluster(id),data=datalong,id="id",
   time="time",event="event",lagvars=c("time","value"),invariant=c("basesize","Group","basenum"),
   maxfu=NULL,first=TRUE,frailty=TRUE)
```

```
mo(20,Liangmo,datalong[is.finite(datalong$size),],weights$iiw.weight[is.finite(datalong$size)],
 singleobs=FALSE,id="id",time="time",keep.first=TRUE,var=FALSE,Yname="size",Xnames="Group",
 Wnames="time",maxfu=NULL,baseline=1)
```

```
## [1] -0.1749
## [1] -0.18
## [1] -0.1389
## [1] -0.1961
## [1] -0.1417
## [1] -0.1666
## [1] -0.131
## [1] -0.1778
## [1] -0.1584
## [1] -0.2142
## [1] -0.1372
## [1] -0.1687
## [1] -0.1844
## [1] -0.152
## [1] -0.175
## [1] -0.1301
## [1] -0.1448
## [1] -0.1353
## [1] -0.1194
## [1] -0.1582
## , , 1
##
##          [,1]
##  [1,] -0.1749
##  [2,] -0.1800
##  [3,] -0.1389
##  [4,] -0.1961
##  [5,] -0.1417
##  [6,] -0.1666
##  [7,] -0.1310
##  [8,] -0.1778
```

```
##  [9,] -0.1584
## [10,] -0.2142
## [11,] -0.1372
## [12,] -0.1687
## [13,] -0.1844
## [14,] -0.1520
## [15,] -0.1750
## [16,] -0.1301
## [17,] -0.1448
## [18,] -0.1353
## [19,] -0.1194
## [20,] -0.1582
```

```
## $est
## [1] -0.1592
```

The mean size of the largest tumour is smaller in the thiotepa group by 0.16cm compared to the placebo group. Note that the Liang method does not provide theoretical standard errors, so these are usually estimated through bootstrapping.

References

Andrews, D.F., and A.M. Herzberg. 1985. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. Springer.

Buzkova, P., and T. Lumley. 2007. “Longitudinal Data Analysis for Generalized Linear Models with Follow-up Dependent on Outcome-Related Variables.” *The Canadian Journal of Statistics* 35: 485–500.

Byar, D., and C. Blackard. 1977. “Comparisons of Placebo, Pyridoxine, and Topical Thiotepa in Preventing Recurrence of State I Bladder Cancer.” *Urology* 10: 556–61.

Follmann, D., M. Proschan, and E. Leifer. 2003. “Multiple Outputation: Inference for Complex Clustered Data by Averaging Analyses from Independent Data.” *Biometrics* 59: 420–29.

Hoffman, E.B., P.K. Sen, and C. Weinberg. 2001. “Within-Cluster Resampling.” *Biometrika* 88: 1121–34.

Liang, Y., W. Lu, and Z. Ying. 2009. “Joint Modeling and Analysis of Longitudinal Data with Informative Observation Times.” *Biometrics* 65: 377–84.

Lin, H., D.O. Scharfstein, and R.A. Rosenheck. 2004. “Analysis of Longitudinal Data with Irregular, Outcome-Dependent Follow-up.” *Journal of the Royal Statistical Society, Series B* 66: 791–813.

Pullenayegum, E.M. 2016. “Multiple Outputation for the Analysis of Longitudinal Data Subject to Irregular Observation.” *Statistics in Medicine* in press.

Pullenayegum, E.M., and L.S.H. Lim. In Press. “Longitudinal Data Subject to Irregular Observation: A Review of Methods with a Focus on Visit Processes, Assumptions, and Study Design.” *Statistical Methods in Medical Research*.