PSpack: A software package for evaluating the causal effect of a longitudinal treatment on an outcome using the method of principal stratification

Constantine E. Frangakis and Ravi Varadhan

Department of Biostatistics, Johns Hopkins University

615 N. Wolfe Street, Baltimore, MD 21205.

January 12, 2003

Introduction

In studies where the primary objective is to evaluate the causal effect of a treatment on an outcome, the method of principal stratification provides a useful approach for properly adjusting for post-treatment variables. In this methodology, the stratification is done with respect to a post-treatment variable and the strata are a cross-classification of subjects defined by the joint potential values of that post-treatment variable. The key property of principal strata is that they are not affected by treatment assignment and hence can be used just as any pre-treatment covariate, such as age. Frangakis and Rubin (2002) provide a thorough discussion of this framework. Recently, Frangakis et al. (2002) evaluated the impact of Baltimore city's needle exchange program (NEP) on HIV seroconversion using this approach. This paper also describes in detail the estimation methodology and the plausible assumptions that are needed to estimate the causal effect. The documentation provided here essentially describes all the steps involved in performing the analysis presented in Frangakis et al. (2002). Even though the documentation is described using the NEP evaluation study as a model, the PSpack package can be used to employ the principal stratification approach to similar longitudinal studies, where the recepit of the treatment and provision of outcome measurement are not directly controlled, but a longitudinal factor is controlled and this factor is associated with both taking the treatment, as well as providing outcome measurement. It is currently capable of modelling binary outcomes, and multiple levels of controlled factor (e.g., distance). It will soon be extended to handle ordinal outcomes (with more than two levels). The user of the software is strongly urged to read the paper by Frangakis et al. (2002) for a better understanding of the assumptions involved in the use of this methodology.

Software requirements

The software is written to be used in a statistical programming environment called R, which is open-source and freely available. To obtain information on how to obtain and install R, check the website: *www.r-project.org*. With a few (possibly) minor changes, it can also be used in an Splus environment. The following files are required for the use of PSpack:

- routines.r - contains some R functions to perform data preparation and other analytic tasks.
- routines.dll (for Windows) - contains the library of executable functions which can be dynamically linked at the beginning of an R session. This enables the execution of several Fortran subroutines which perform more intensive computing tasks.
- routines.so (for Unix) - same purpose as routines.dll, but for Unix users.
- routines.f - the Fortran source codes for creating the .dll or .so file are contained in this file. This file is not necessary, unless the user wants to make some changes and recompile to create a new library.
- sample.r - contains a sample session illustrating the use of PSpack for the NEP study.
- datawide - contains data, as an R object, for the sample session.

These required files are archived in “pspack.zip” and “pspack.tar.gz” for the Windows and Unix users, respectively.

Description of the basic data

The basic data object required for the analysis is a dataframe where each row contains information for a subject. This information includes (in the specified order): subject id, fixed (not varying with time) covariates, time-varying covariates, controlled factor (time-varying), treatment/exposure (time-varying), and outcome (time-varying)., and censoring indicator (time-varying). The time-varying variables are measured at each visit. For example, in the NEP study, measurements are taken every semester. There are 5 fixed covariates, the first 5 principal components determined from a set of 24 baseline characteristics of the subjects. There are no time-varying covariates. However, if there were any, they should be entered one after the other, as follows:

$var1.1, \cdots, var1.k, var2.1, \cdots, var2.k, \cdots,$

where there are k repeated measurements or visits and $var1.1$ denotes variable 1 measured at first visit, etc.. In the NEP study, there are 12 semesters at which information is available (k=12). The controlled factor is the distance (D) to the closest NEP site from the subject’s domicile. Since the NEP sites and/or the subjects’ domicile can change with time, the distance also varies with time. The exposure variable (E) is binary indicating whether the subject exchanged needles at the NEP, at least once, in the previous semester (0 = No; 1 = Yes). The outcome variable (Y) is binary indicating whether the subject tested positive for HIV antibodies at the current visit (0 = No; 1 = Yes). If a subject drops-out at a particular visit, then the outcome is designated as “NA” at that visit.

In the NEP example, there are 1170 subjects. Each subject has a unique ID. The five fixed covariates are represented by a 1170×5 matrix. There are no time-dependent covariates. Time ranges from 1 to 12, where each unit is a semester. The controlled factor, distance, is time-varying and hence can be represented by a 1170×12 matrix. Here, distance, which is continuous, is dichotomized as “near” and “far”, but in general it can be a factor variable with any number of levels. Exposure, outcome, and censoring indicators can also be thought of 1170×12 matrices, but as stated earlier it should be noted that once the outcome becomes “NA”, the rest (after that time) of the information for that subject is irrelevant, and is ignored. Thus the entire data frame, containing all the data, has 1170 rows and 42 (=1+5+12*0+12*3) columns. This is the “wide” format. This data frame, let us call it *datawide* is the primary input for the analysis.

Creating the “long” data

The function “makebig” takes *datawide* as input and produces as output another data frame which is in the “long” format. Let us call this *databig*. This data frame has *tatrisk* rows for each subject, corresponding to the time at risk for that subject. Thus, it has *ntatrisk* total number of rows, which is the sum of all the times at risk for each subject. For example, a subject who tests HIV positive at the 5th semester will have 5 rows and a subject who drops out at the 9th semester before seroconverting will have 9 rows. The *databig* object also has 11 (=1+5+0+1+4) columns, a column for ID, 5 columns for fixed covariates (the value for each covariate is simply repeated 12 times for each subject), 1 column for time (1 to 12), and 4 columns - one each for distance, exposure, outcome, and censoring indicator. The censoring variable, which is newly created, is also binary indicating whether the subject provided outcome measurement at the current visit (0 = Yes; 1 = No). A subject’s censoring indicator assumes a value of 1 at a particular visit, if and only if he/she provided measurements at the previous visit but dropped out after that, i.e. his outcome is “NA”. It is also assumed that once a subject drops out of the study, he/she does not get back into it. In addition to these, it may be useful to have lagged variables for the longitudinal controlled factor, here distance, and for the exposure variable. These lagged variables may generally be included in the models for principal stratum, outcome, and censoring. There is a function called “makelagvars” to create these variables.

Creating the “full” data

The final data object that needs to be created is called *datafull*. It will be required by the EM algorithm to estimate all the model parameters. As mentioned in the Introduction, the principal stratification for the NEP study is based on a post-treatment variable. Here the post-treatment variable used for stratification describes how a subject behaves in terms of exchanging needles at near versus far distances to the NEP site. This can be generally thought of as encoding subject’s propensity to exchange, as well as some characteristics of NEP program. It is assumed that subjects who exchange when the NEP is placed farther from their residence, will also do so when the NEP is located closer, and conversely, if they don’t exchange at a nearer distance, they won’t do so at a farther distance. Based on this monotonicity assumption, there are 3 principal strata, when the distance is dichotomized as near and far. In the first strata (Strata 1) are the subjects who will exchange at far and near distances. In the third strata (Strata 3) are the subjects who will not exchange regardless of the NEP’s location. In the intermediate strata (Strata 2) are the subjects whose behavior is affected by the location of the NEP. They will exchange only when the NEP is placed nearer.

Now, the *datafull* is created by the function “makefull” as follows. For each subject-time unit, based on the observed distance and exchange behaviour, we determine the possible principal strata. For example, if for subject i at time j, we have $D_{ij} = 0$ (near) and

$E_{ij} = 1$ (exchanged) , then the possible principal strata S_{ij} are 1 and 2. This information will be reflected in *datafull* by having the corresponding row from the *databig* object repeated twice, and by having an additional column for denoting the principal stratum, S. Therefore, *datafull* will have more rows than *databig*. The function “makefull”, in addition to creating *datafull* also provides a vector of principal strata values, one for each subject-time unit, which is a random draw from the allowable values. This is used later by the function “EMstart” to determine a good starting point to the EM algorithm for the parameters of the principal strata model .

Model specification

The user has to define three different models in order to be able to estimate the causal effect of the treatment. The first model is a proportional-odds logistic regression model for the principal strata, since the principal strata are ordered from 1 to 3 (for the NEP example). The principal strata model is generally specified in terms of fixed covariates, time, and past history of the subject (this may include previous time-varying covariates, distances, and exchange behavior). The second model is a binary logistic regression model for the censoring indicator (0 = No; 1 = Yes). This model can generally be specified in terms of fixed covariates, time, past history, current exchange behavior, and the principal strata. The final model is also a binary logistic regression model for the outcome variable (0 = No; 1 = Yes). This model can also generally be specified in terms of fixed covariates, time, past history, current exchange behavior, and the principal strata. The causal effect of interest is the coefficient corresponding to the current exchange behavior. It should be noted that all these models are written for subject-time units, i.e. the principal strata, censoring indicator, and outcome are defined for the unit (i,t) - person i and semester t. Another implicit notion is that the models are defined only for subjects who are still at risk at semester t, i.e. those who haven't dropped out and haven't seroconverted at time t.

The model formulas are written in the Wilkinson-Rogers notation, which is the convention followed in R and Splus. For example, the model formulas for the three models may be written as follows:

```
S ~ X1 + Em1 + Dm1 + time + E
C ~ X1 + Em1 + Dm1 + time + S + E
Y ~ X1 + Em1 + Dm1 + time + S + E
```

where S,C, and Y are the principal stratum, censoring indicator, and outcome, respectively, and X is a fixed covariate, Dm1 and Em1 are the distance and exchange behavior observed at the previous visit, and E is the current exchange behavior.

EM algorithm to estimate the model parameters

The observed data for a person i at time t are (given that he/she is at risk for outcome at t), past history (fixed covariates, previous values of time-varying covariates, previous distances and exchange behavior), current distance, exchange behavior, censoring indicator at time t, and outcome (if not censored at t). The unobserved (latent) data is the information on the principal stratum to which a unit, (i,t), belongs. The observed data together and the latent principal strata together make up the complete data. The goal is to estimate the parameters of the three models, which are optimal in the sense that they maximize the likelihood of obtaining the observed data. This, however, is not directly feasible since the observed data likelihood depends on the unknown principal strata. This difficulty can be overcome by the use of the EM algorithm. The EM algorithm works as follows: (i) in E-step, evaluate the expectation of log-likelihood of the complete data, where the expectation is with respect to the conditional density of latent data given the observed data, (ii) in the M-step, the expectation from (i), which is a function of the parameters involved in the the latent data and observed data models, is maximized with respect to these parameters.

In PSpack, the E-step computations are performed by a call to a Fortran subroutine. Because of the additivity of the contributions from the three models to the expectation of complete data log-likelihood, the M-step is performed by three separate maximizations. The maximization of the principal strata model component is achieved via a call to an R function called *polr*, which fits a proprortional-odds logistic regressio model. The maximization of the censoring and outcome model components are achieved via separate calls to the *glm*, which fits a logistic model, with logit link. The function “EM.ps” performs the EM computations.

The EM algorithm is an iterative method. Good starting values for the parameters are essential for both faster convergence as well as convergence to a global maximum. In PSpack, good starting values are obtained by assigning equal probabilities (weights) to each allowable principal stratum for a unit (i,j), i.e. the conditional distribution of the principal strata for a unit (i,j) given observed data, is assumed to be uniform over all the allowable strata. The function “EMstart” computes the starting values.

Calculation of the covariance matrix

The EM algorithm, unfortunately, does not provide an estimate of the covariance matrix of the parameters. The covariance matrix is the inverse of the Hessian matrix, which is a matrix of second-derivatives of the negative log-likelihood, evaluated at the parameter values to which the EM algorithm converged. The Hessian matrix is computed in PSpack by means of a call to a Fortran subroutine, it is then inverted to obtain the covariance matrix.

A sample session

```
# First, get all the necessary R functions
> source("routines.r")
# Load into memory, the library of Fortran functions executable in R
> dyn.load("routines.dll")  # In Unix:  dyn.load("routines.so")
# Make the MASS library available; this is needed to run ‘‘polr’’
> library(MASS)
# Seeding the random number generation
> rngseed(12345) # you can use any integer; to reproduce results exactly, use the same seed

# Get an example data set in the wide-format
> datawide <- dget("datawide")
# Define the # of fixed and time-dependent covariates
> nfix <- 5
> nvary <- 0
# Define the time units
> times <- c(1:12)

# Step 1. get big object, databig, with columns: (id,fixed X’s,time-dependent Z’s,time,D,E,Y,C)
#
# Input: datawide with columns: (id,fixed X’s,Z1.1,Z1.2,...,Z1.K,Z2.1,...,Zp.K,D.1,...,D.K,
# E.1,...,E.K,Y.1,...,Y.K)
> bigobj <- makebig(nfix,nvary,time=times,data=datawide)
> databig <- bigobj$databig
# Set the outcome value, at which drop-out occurs, to an arbitraray negative number
# This is required since Fortran can’t deal with "NA"
> databig$Y[databig$C==1] <- -999.

# Step 2. Create auxiliary variables, such as lagged distance and exposure (lag=1),
#  needed for defining models. Add these variables to databig.
# i.e., Dm1(i,t) = D(i,t-1), and Em1(i,t) = E(i,t-1).
# Inputs: databig, idcol, dcol, ecol
> idcol <- 1  # column no. of the subject ID variable in databig
> dcol <- as.integer(1+nfix+nvary+2)  # column no. of Distance variable in databig
> ecol <- as.integer(dcol+1)   # column no. of Exchange variable in databig

> lagvar <- makelagvars(data=databig,id=idcol,dcol=dcol,ecol=ecol)
> databig$Dm1 <- lagvar$Dlag
> databig$Em1 <- lagvar$Elag

# Step 3. make the "full" object, datafull.
# Inputs: databig, dcol, ecol
# In additional to dataful, other outputs are:
# sbig: a random draw from the allowable principal strata for each subject-time unit
# nsbig: number of allowable principal strata for each subject-time unit

> fullobj <- makefull(data=databig,dcol=dcol,ecol=ecol)
> datafull <- fullobj$datafull
> sbig <- fullobj$sbig
> nsbig <- fullobj$nsbig
```

```
# Step 4. define formulas and design matrices in full dimensions.
# INPUT: model formulas for principal strata (S), censoring indicator (C), and outcome (Y),

> c.fmla <- as.formula("C ˜ X1 + Em1 + Dm1 + time + S + E")
> s.fmla <- as.formula("as.factor(S) ˜ X1 + Em1 + Dm1 + time")
> y.fmla <- as.formula("Y ˜ X1 + Em1 + Dm1 + time + S + E")

# Step 5. Obtain good initial starting values for the EM algorithm
# Input: formulas, sbig, databig

theta.0 <- EMstart(smodel=s.fmla,ymodel=y.fmla,cmodel=c.fmla,sbig,data=databig)

# Step 6. Run the EM algorithm until convergence
# Input: formulas, sbig, theta.0, nsbig, datafull, rel.tol(convergence criterion), and
#   maxiter(maximum number of iterations).
# Ouputs:  Converged parameter values and hessian matrix

> ans <- EM.ps(smodel=s.fmla,ymodel=y.fmla,cmodel=c.fmla,theta.0,nsbig,data=datafull,
+   rel.tol=1.e-07, maxiter=100)
> theta <- ans$par  # converged parameter estimates
> theta[length(theta)]  # the causal effect parameter
> se <- sqrt(diag(ans$vcov))  # estimated standard errors of parameter estimates
> se[length(theta)]  # std. error of the causal effect
```