Survival Analysis for Cohorts with Missing Covariate Information

NestedCohort fits Kaplan-Meier and Cox Models to estimate standardized survival and attributable risks for studies where covariates of interest are observed on only a sample of the cohort.

by Hormuzd A. Katki and Steven D. Mark

Introduction

Large epidemiologic cohort studies are often designed to allow researchers to conduct many studies involving different exposures and survival outcomes. Often, all survival outcomes and certain easily-ascertainable covariates are observed on everyone. However, the exposures of interest to each particular study nested within the cohort are typically observed on only a sample of the cohort. This is done because when a big percentage of outcomes are censored (as is the case for a rare disease or for a cohort with little follow-up time), measuring the exposure on only a sample of the censored sacrifices little statistical efficiency but can save a lot of money, effort, and precious biosamples. A study nested within a cohort is an example of a two-phase study where the first phase is a cohort. Typical examples are case-control studies conducted within defined cohorts, nested case-control, or case-cohort studies.

However, commonly-available software for analysis is limited in many ways, most notably, by only estimating relative risks. **NestedCohort** implements much of the methodology of Mark and Katki (2006) to extend current software options for studies nested within cohorts:

1. **NestedCohort** estimates not just relative risks, but also absolute and attributable risks. **NestedCohort** can estimate non-parametric Kaplan-Meier survival curves for each level of the exposures and also fit Cox models to estimate survival and attributable risks that are standardized for confounders.

2. **NestedCohort** allows cases to have missing exposures. Standard nested case-control and case-cohort software can suffer from bias if cases are missing exposures.

3. **NestedCohort** allows stratified sampling of subjects for whom the exposure is observed on, stratifying on any variables, including outcome time, available on all cohort members. This allows for frequency matching on confounders, or oversampling on the extremes of surrogates for exposures to improve efficiency.

4. **NestedCohort** can extract efficiency out of auxiliary variables available on all cohort members. For example, no variable on the causal pathway between exposure and outcome, nor any variable correlated with the missing exposures (e.g. "surrogates" for exposure), can be used as a covariate in a Cox model. However, these auxiliary variables can be used by the sampling model (described below) to improve efficiency in the risk estimates.

5. **NestedCohort** relies on a sampling model to estimate the probability of each person having his exposure observed, and weights estimating equations by the inverse of these probabilities. Using weights estimated from a correctly specified sampling model greatly increases the efficency of the risk estimates vs. using the 'true' weights (see Mark and Katki (2006)).

6. The exposure need not be a scalar, but can be a vector of exposures. For relative risks, all covariates and exposures can be continuous or categorical.

7. Even if you are only interested in relative risks, **NestedCohort** can more effciently estimate relative risks than standard case-control analyses, because **NestedCohort** uses subjects with the missing exposures (who provide information through the outcome and other covariates observed on them), and can exploit auxiliary variables to increase efficiency.

NestedCohort does not currently support fine matching on variables or time.

NestedCohort has three functions that we demonstrate in this article:

1. `nested.km`: Estimates the Kaplan-Meier survival curve for each level of categorical exposures.

2. `nested.coxph`: Fits the Cox model to estimate relative risks. All covariates and exposures can be continuous or categorical.

3. `nested.stdsurv`: Fits the Cox model to estimate standardized survival probabilities, and Population Attributable Risk (PAR). All covariates and exposures must be categorical.

Example Study Nested in a Cohort

In our example, Abnet et al. (2005) observe esophageal cancer survival outcomes and relevant

confounders on the entire cohort. We are interested in the effect of concentrations of various metals, especially zinc, on esophageal cancer. However, measuring metal concentrations consumes precious esophageal biopsy tissue and requires a costly measurement technique. Thus we measured concentrations of zinc (as well as iron, nickel, copper, calcium, and sulphur) on a sample of the cohort. This sample oversampled the cases and those with advanced baseline histologies (i.e. those most likely to become cases) since these are the most informative subjects. Due to cost and availability constraints, less than 30% of the cohort could be sampled. Although this study has no auxiliary variables, we show how to use them if they were available. For this example, **NestedCohort** will provide adjusted hazard ratios, standardized survival probabilities, and PAR for the effect of zinc on esophageal cancer.

Specifying the Sampling Model

NestedCohort requires you to specify the variables that determine the sampling scheme for whom the missing exposure is observed on. These variables account for the sampling scheme by estimating the probability that each subject would would have their exposure observed on them. By default, this sampling probability is modeled with a logistic regression of sampling status on the sampling variables. The inverse of these estimated sampling probabilities are used to weight each observation in the estimation of the survival curves. For details, see Mark and Katki (2006).

To choose the sampling variables, note that any variable that has information about the outcome or missing exposures is potentially worthwhile to sample on (Mark and Katki (2006)). Sampling on case/control status is almost always important: you should try to observe the exposure on as many cases as possible. For the zinc data, baseline esophageal histology is a powerful potential confounder, as it is tightly linked to being a case, and if lack of zinc causes esophageal cancer, then it may well cause precancerous lesions. To control for baseline histology, we chose to frequency match on it. Here is our sampling scheme:

Baseline Histology	Case	Control	Total
Normal	14 / 22	17 / 221	31 / 243
Esophagitis	19 / 26	22 / 82	41 / 108
Mild Dysplasia	12 / 17	19 / 35	31 / 52
Moderate Dysplasia	3 / 7	4 / 6	7 / 13
Severe Dysplasia	5 / 6	3 / 4	8 / 10
Carcinoma In Situ	2 / 2	0 / 0	2 / 2
Unknown	1 / 1	2 / 2	3 / 3
Total	56 / 81	67 / 350	123 / 431

For each cell, the number to the right of the slash is the total cohort members in that cell, the left is the number we sampled to have zinc observed (i.e. in the top left cell, we measured zinc on 14 of the 22 members who became cases and had normal histology at baseline). Note that for each histology, we sampled roughly 1:1 cases to controls (frequency matching), and we oversampled the more severe histologies (who are more informative since they are more likely to become cases). 30% of the cases could not be sampled.

This non-representative sampling will be accounted with inverse-probability weights by the sampling model. Since there are 7 baseline histologies, and case/control status, then the sampling probability for each subject depends on which of 14 strata they belong to. We estimated the sampling fractions using a logistic model regressing having zinc measurements on the 14 strata, allowing each stratum its own sampling fraction. To do this, each function will use the statement `samplingmod="ec01*basehist"`.

To be practical, the sampling design should not be so complex that it cannot be carried out. Also, the more sampling strata you choose, the more likely that an observation will get a zero probability of having their exposure observed. Every non-empty stratum must have have someone sampled in it, or **NestedCohort** will not work. To insure that there is some sample in each stratum, you may have to collapse strata. Also, if the sampling is not under your direct control, then it is important that the sampling model contain all covariates that could potentially affect whether the exposures are measured on any subject. For making valid estimates, **NestedCohort** depends on the sampling model containing all variables used in the sampling scheme. Finally, **NestedCohort** does not support fine matching on variables or time. However, you can always include any frequency-matched variables as covariates in the analysis to further control for their effects.

Formally, missingness should not allowed for any variable in the sampling model. However, if there is missingness, for convenience, **NestedCohort** will remove from the cohort any observations that have missingness in the sampling variables and will print a warning to the user. There should not be too many such observations.

Kaplan-Meier Curves

You make non-parametric (Kaplan-Meier) survival curves by quartile of zinc level using `nested.km`. These Kaplan-Meier curves have the usual interpretation: they do not standardize for other variables, and do not account for competing risks.

To use this, you must provide both a legal formula as per the `survfit` function and also a sampling model to calculate stratum-specific sampling fractions. Note that the '`survfitformula`' and

'samplingmod' require their arguments to be inside double quotes. The 'data' argument is required, you must provide the data frame within which all variables reside in. This outputs the Kaplan-Meier curves into a survfit object, so all the methods that are already there to manipulate survfit objects can be used[1].

To examine survival from cancer within each quartile of zinc, and allowing different sampling probabilities for each of the 14 strata above, use nested.km, which prints out a table of risk differences versus the level named in 'exposureofinterest'; in this case, it's towards "Q4" which labels the 4th quartile of zinc concentration:

```
> library(NestedCohort)
> mod <- nested.km(survfitformula =
+          "Surv(futime01,ec01==1)~znquartiles",
+          samplingmod = "ec01*basehist",
+          exposureofinterest = "Q4", data = zinc)

Risk Differences vs. znquartiles=Q4 by time 5893
        Risk Difference  StdErr       95% CI
Q4 - Q1         0.28175 0.10416  0.07760 0.4859
Q4 - Q2         0.05551 0.07566 -0.09278 0.2038
Q4 - Q3         0.10681 0.08074 -0.05143 0.2651

> summary(mod)
[...]

308 observations deleted due to missing
                znquartiles=Q1
 time n.risk n.event survival std.err    95% CI
  163  125.5    1.37    0.989  0.0108 0.925 0.998
 1003  120.4    1.57    0.976  0.0169 0.906 0.994
 1036  118.8    1.00    0.968  0.0191 0.899 0.990
[...]

                znquartiles=Q2
 time n.risk n.event survival std.err    95% CI
 1038  116.9    1.57    0.987  0.0133 0.909 0.998
 1064  115.3    4.51    0.949  0.0260 0.864 0.981
 1070  110.8    2.33    0.929  0.0324 0.830 0.971
[...]
```

summary gives the lifetable. Although summary prints how many observations were "deleted" because of missing exposures, the "deleted" observations still contribute to the final estimates via estimation of the sampling probabilities. Note that the lifetable contains the weighted numbers of those at risk and who had the developed cancer.

The option 'outputsamplingmod' allows you to return the sampling model that the sampling probabilities were calculated from. Examine this model if you're warned that it didn't converge. If 'outputsamplingmod=T', then nested.km will output a list with 2 components, the survmod component being the Kaplan-Meier survfit object, and the other samplingmod component being the sampling model.

Plotting Kaplan-Meier Curves

You can make Kaplan-Meier plots with the plot function for survfit objects. All plot options for survfit objects can be used.

```
> plot(mod,ymin=.6,xlab="time",ylab="survival",
+      main="Survival by Quartile of Zinc",
+      legend.text=c("Q1","Q2","Q3","Q4"),
+      lty=1:4,legend.pos=c(2000,.7))
```

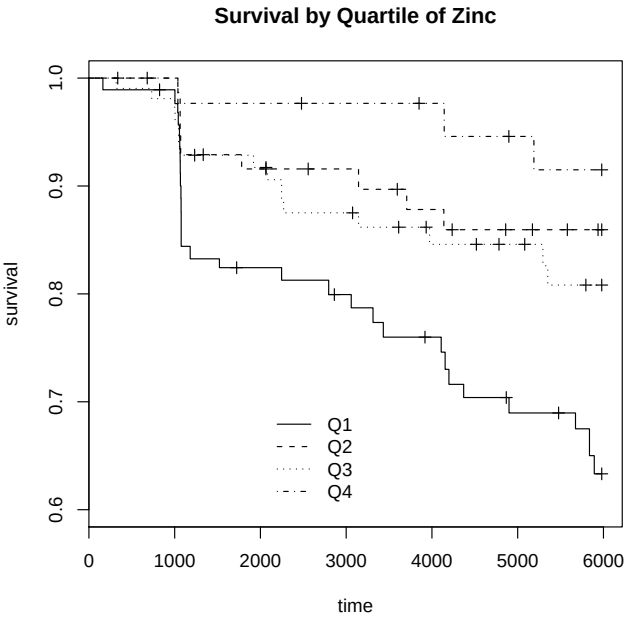

Figure 1: Kaplan-Meier plots by nested.km().

nested.km has some requirements:

1. All variables are in a dataframe denoted by the 'data' argument.
2. No variable in the dataframe can be named o.b.s.e.r.v.e.d. or p.i.h.a.t.
3. 'survfitformula' must be a valid formula for survfit objects: All variables must be factors.
4. Does not support staggered entry into the cohort. The survival estimates will be correct, but their standard errors will be wrong.

Cox Models: Relative Risks

To fit the Cox model, use nested.coxph, which relies on the function coxph that is already in the survival package, and imitates its syntax as much as possible. In this example, we are interested in estimating the effect of zinc (as zncent, a continuous variable standardized to 0 median and where a 1 unit change is an

[1] nested.km uses the weights option in survfit to estimate the survival curve. However, the standard errors reported by survfit are usually quite different from, and usually much smaller than, the correct ones as reported by nested.km.

increase of 1 quartile in zinc) on esophageal cancer, while controlling for sex, age (as agepill, a continuous variable), smoking, drinking (both ever/never), baseline histology, and family history (yes/no). We use the same sampling model ec01*basehist as before. The output is edited for space:

```
> coxmod <- nested.coxph(coxformula =
+   "Surv(futime01,ec01==1)~sex+agepill+basehist+
     anyhist+zncent",
+   samplingmod = "ec01*basehist", data = zinc)

> summary(coxmod)
[...]
                          exp(coef) lower/upper.95
sexMale                        0.83    0.38   1.79
agepill                        1.04    0.99   1.10
basehistEsophagitis            2.97    1.41   6.26
basehistMild Dysplasia         4.88    2.19  10.88
basehistModerate Dysplasia     6.95    2.63  18.38
basehistSevere Dysplasia      11.05    3.37  36.19
basehistNOS                    3.03    0.29  30.93
basehistCIS                   34.43   10.33 114.69
anyhistFamily History          1.32    0.61   2.83
zncent                         0.73    0.57   0.93

[...]
Wald test = 97.5  on 10 df,   p=2.22e-16
```

This is the exact same coxph output, except that the R^2, overall likelihood ratio and overall score tests are not computed. The overall Wald test is correctly computed.

nested.coxph has the following requirements:

1. All variables are in the dataframe in the 'data' argument.
2. No variable in the dataframe can be named o.b.s.e.r.v.e.d. or p.i.h.a.t.
3. Must use Breslow tie-breaking.
4. No 'cluster' statements allowed.

However, nested.coxph does allow staggered entry into the cohort, stratification of the baselize hazard via 'strata', and use of 'offset' arguments to coxph (see help for coxph for more information).

Standardized Survival and Attributable Risks

nested.stdsurv first estimates hazard ratios exactly like nested.coxph, and then also estimates survival probabilities for each exposure level as well as PAR for a given exposure level, standardizing both for confounders. To standardize, the formula for a Cox model must be split in two pieces: the argument 'exposures' denotes the part of the formula for the exposures of interest, and 'confounders' which denotes the part of the formula for the confounders. All variables in either part of the formula must be factors. In either part, do not use '*' to mean interaction, use interaction.

In the zinc example, the exposures are 'exposures="znquartiles"', a factor variable denoting which quartile of zinc each measurement is in. The confounders are 'confounders="sex+agestr+basehist+anyhist"', these are the same confounders in the hazard ratio example, except that we must categorize age as the factor agestr. 'timeofinterest' denotes the time at which survival probabilities and PAR are to be calculated at, the default is at the last event time. 'exposureofinterest' is the name of the exposure level to which the population is to be set at for computing PAR; 'exposureofinterest="Q4"' denotes that we want PAR if we could move the entire population's zinc levels into the fourth quartile of the current zinc levels. 'plot' plots the standardized survivals with 95% confidence bounds at 'timeofinterest' and returns the data used to make the plot. The output is edited for space:

```
> mod <- nested.stdsurv(outcome =
+         "Surv(futime01,ec01==1)",
+         exposures = "znquartiles",
+         confounders = "sex+agestr+basehist+anyhist",
+         samplingmod = "ec01*basehist",
+         exposureofinterest = "Q4", plot = T, main =
+         "Time to Esophageal Cancer by Quar-
tiles of Zinc",
+         data = zinc)

Std Survival for znquartiles by time 5893
      Survival  StdErr 95% CI Left 95% CI Right
Q1      0.5054 0.06936      0.3634       0.6312
Q2      0.7298 0.07768      0.5429       0.8501
Q3      0.6743 0.07402      0.5065       0.7959
Q4      0.9025 0.05262      0.7316       0.9669
Crude   0.7783 0.02283      0.7296       0.8194

Std Risk Differences vs. znquartiles = Q4 by time 5893
           Risk Difference  StdErr       95% CI
Q4 - Q1             0.3972 0.09008  0.22060 0.5737
Q4 - Q2             0.1727 0.09603 -0.01557 0.3609
Q4 - Q3             0.2282 0.08940  0.05294 0.4034
Q4 - Crude          0.1242 0.05405  0.01823 0.2301

PAR if everyone had znquartiles = Q4
    Estimate StdErr 95% PAR CI Left 95% PAR CI Right
PAR   0.5602 0.2347         -0.2519           0.8455
```

The first table shows the survivals for each quartile of zinc that are standardized for all the confounders, as well as the 'crude' survival, which is the observed survival in the population (so is not standardized). The next table shows the standardized survival differences vs. the exposure of interest. The last table shows the PAR, and the CI for PAR is based on the log(1-PAR) transformation (this is often very different from, and superior to, the naive CI without transformation). summary(mod) would yield the

same hazard ratio output as if the model had been run under `nested.coxph`.

The plot is in figure 2. This plots survival curves; to plot cumulative incidence (1-survival), use `cuminc=T`. The 95% CI bars are plotted at `timeofinterest`. You can use any `plot` options: e.g. '`main`' to title the plot.

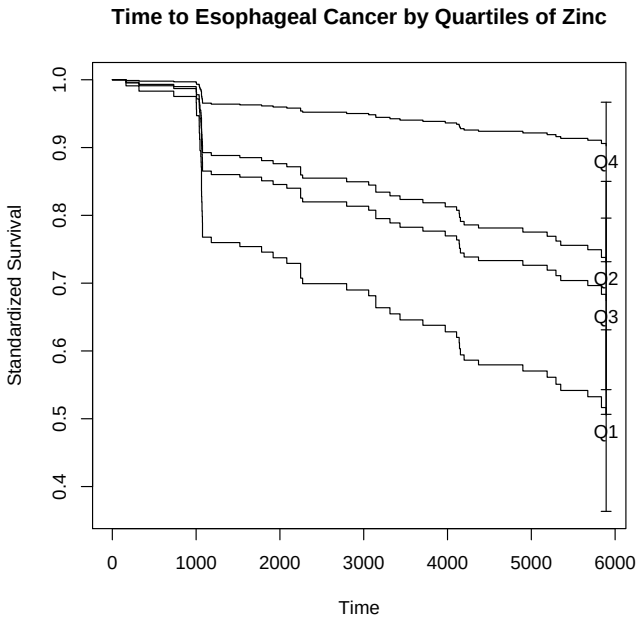

Figure 2: Survival curves for each zinc quantile, standardized for confounders

`nested.stdsurv` has some requirements:

1. All variables are in the dataframe in the '`data`' argument.
2. No variable in the dataframe can be named `o.b.s.e.r.v.e.d.` or `p.i.h.a.t.`
3. The variables in the '`exposures`' and '`confounders`' must be factors, even if they are binary. In these formulas, never use '*' to mean interaction, use `interaction`.
4. Does not support staggered entry into the cohort.
5. Does not support the baseline hazard to be stratified. '`cluser`' and '`offset`' arguments are not supported either.
6. Only allows Breslow tie-breaking.

Auxiliary variables

Auxiliary variables observed on the entire cohort cannot be used in the risk equation. For example, auxiliaries might be surrogates for the exposure (for example, you may have a rough measure of zinc available on everyone) or on a causal pathway between exposure and outcome (for example, in studies of genetic polymorphisms and disease, family history is a variable on a causal pathway from polymorphism to disease). You cannot include auxiliaries as covariates in the risk regression because they could distort the association between exposure and disease, adjusted for confounders. However, information can be extracted out of auxiliaries by incorporating them in the sampling model.

Although the zinc dataset does not have any auxiliary variables, let's pretend we have a categorical surrogate named `znauxiliary` observed on the full cohort. You could sample based on `znauxiliary` to get as wide a zinc distribution possible and thus improve efficiency. Clearly, you would then include `znauxiliary` as a sampling variable in the sampling model with `samplingmod="ec01*basehist*znauxiliary"`

Even if you don't choose to sample based on `znauxiliary`, you can still include `znauxiliary` in the sampling model as above. This is because even though you don't explicitly sample on it, if `znauxiliary` has something to do with zinc, and zinc has something to do with either `ec01` or `basehist`, you are implicitly sampling on `znauxiliary`. The simulations in Mark and Katki (2006) show the efficiency gain from including auxiliary variables in the sampling model. Including auxiliary variables will always reduce the standard errors of the risk estimates.

Multiple exposures

Multiple exposures (with missing values) are included in the risk regression just like any other variable. For example, if we want to estimate the esophageal cancer risk from zinc and calcium jointly, the Cox model would include `cacent` as a covariate. If you cut calcium into quartiles into the variable `caquartiles`, you can include it as an exposure with `nested.stdsurv` with `exposures="znquartiles+caquartiles"`.

With multiple exposures, the key is to make sure that the missingness pattern is monotone. In a monotone pattern, the exposures with missing values are entirely observed for some people, and on others are entirely unobserved. This is the case in this study, as all the metals were all measured simultaneously on the same subjects and none were measured on the others. NestedCohort is designed to work for monotone missingness. In a non-monotone pattern, some subjects would have zinc observed but not calcium, and vice versa, with few having both observed. NestedCohort would treat the subjects who have both measurements as the complete data, and all the others as missing data, and will work, albeit with potential efficiency loss since so much

data is considered missing. If there is serious non-monotonicity, a different technique, like multiple imputation, may be better to use.

Full cohort analysis

If all covariates are observed on the full cohort, you can still use NestedCohort to estimate the standardized survival and attributable risks, by setting `samplingModel="1"`, tol force equal weights for all cohort members. `nested.km` will work exactly as `survfit` does. The Cox model standard errors will be those you get from `coxph` with `robust=T`.

Bibliography

Abnet, C. C., Lai, B., Qiao, Y.-L., Vogt, S., Luo, X.-M., Taylor, P. R., Dong, Z.-W., Mark, S. D., and Dawsey, S. M. (2005). Zinc concentration in esophageal biopsies measured by x-ray fluorescence and cancer risk. *Journal of the National Cancer Institute*, 97(4):301–306.

Mark, S. D. and Katki, H. A. (2006). Specifying and implementing nonparametric and semiparametric survival estimators in two-stage (sampled) cohort studies with missing case data. *Journal of the American Statistical Association*, 101(474):460–471.

Hormuzd A. Katki
Division of Cancer Epidemiology and Genetics
National Cancer Institute, NIH, DHHS, USA
`katkih@mail.nih.gov`

Steven D. Mark
Department of Preventive Medicine and Biometrics
University of Colorado Health Sciences Center
`Steven.Mark@UCHSC.edu`