The small-area case time series and space-time-stratified case crossover design

A case study for applications in environmental epidemiology

Yuantong (Anthony) Sun

24 February 2023

Contents

In the individual CTS design, cases were represented by subjects. But the design can be extended by defining observational units as small geographical areas Antonio mentioned in the recent published paper.

Yuming Guo's research group also extended the time-stratified case crossover study to space-time-stratified case crossover study to accommodate the increasing demand for multilocation study. the link to Guo's paper

When the aggregated CTS adjust for temporal variation conditional on year, month and day of week and for spatial variation conditional on small areas, both methods will provide the same effect estimates if there are only exposure data at small-area level. (The conclusion was drawn from Guo's paper and extended by Anthony)

Preparation

The following packages are loaded in the session, and need to be installed to run the R code:

```
library(dlnm) ; library(gnm) ; library(data.table) ; library(splines); library(dplyr);
library(tidyr); library(dlnm); library(splines); library(gnm); library(survival);
library(truncnorm); library(lubridate); library(stringr); library(scales)
```

The data used in this case study are generated from Antonio Gasparrini's paper - "A tutorial on the case time series design for small-area analysis". The use of data follows GNU public licence, which permits commercial use, modification, distribution, patent use and private use.

Load the data

```
london <- readRDS("data/london_mortality.rds")
```

Now we have the London mortality dataset, reporting the deaths that occured in London in summertime (June to August) in 2006 and 2013. The data is a time series across 983 middle layer super output areas (MSOA). For column “dtot”, dtot (total deaths) = d074 (deaths 0-74 years old) + d75plus (deaths 75+ years old).

Statistical analysis with small-area CTS design

First, check the completeness of this time series dataset

```
# DEFINE SERIES OF UNIQUE MSOA AND DATES
seqmsoa <- sort(unique(london$MSOA11CD))
seqdate <- sort(unique(london$date)) # 92 days/year * 2 years = 184 days

# COMPLETE TIME SERIES
ts <- expand.grid(county_fips=seqmsoa, date=seqdate) |>
  data.table()
# The original dataset has the same columns with complete time series [PASS]

# ORDER (IMPORTANT FOR KEEPING THE TIME SERIES SEQUENCE BY MSOA)
setkey(london, MSOA11CD , date)
```

Fit the conditional Poisson model

```
# DEFINE SPLINES OF DAY OF THE YEAR
spldoy <- onebasis(london$doy, "ns", df=3)

# DEFINE THE CROSS-BASIS FOR TEMPERATURE FROM THE EXPOSURE HISTORY MATRIX
# NB: USE group TO IDENTIFY LACK OF CONTINUITY IN SERIES BY COUNTY AND YEAR
argvar <- list(fun="ns", knots=quantile(london$tmean, c(50,90)/100, na.rm=T))
arglag <- list(fun="ns", knots=1)
group <- factor(paste(london$MSOA11CD, london$year, sep="-"))
cbtmean <- crossbasis(london$tmean, lag=3, argvar=argvar, arglag=arglag,
  group=group)
# To deal with the discontinuation and lag structure of time series,
# we use group feature to address this issue.
summary(cbtmean)
```

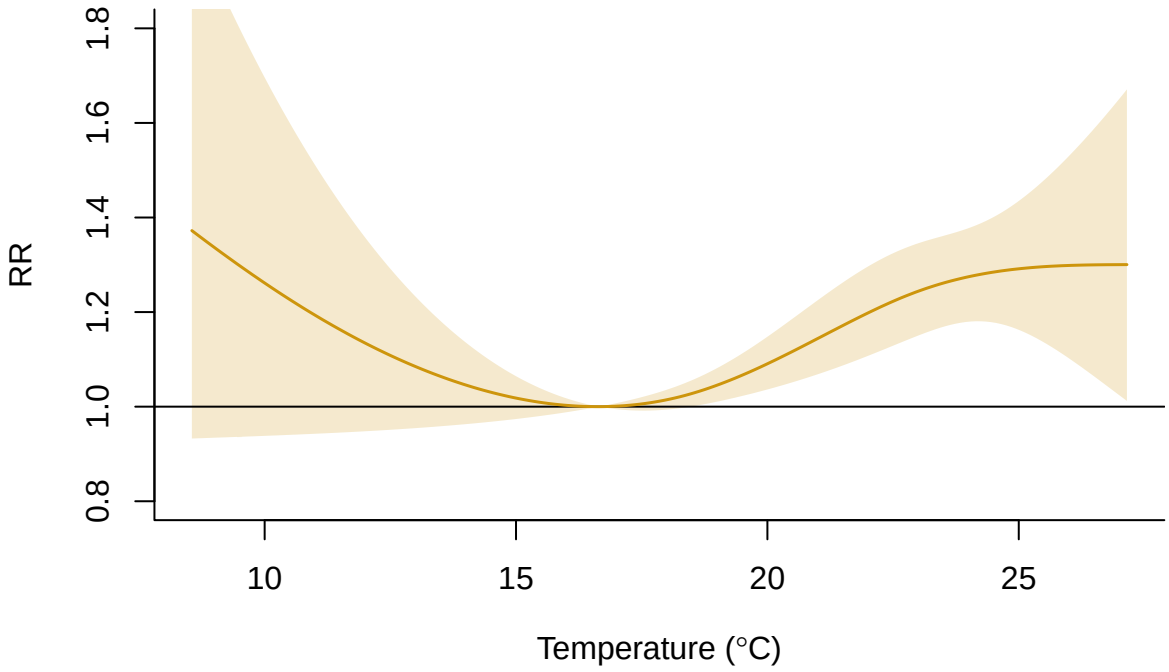
```
## CROSSBASIS FUNCTIONS
## observations: 180872
## groups: 1966
## range: 8.551893 to 27.17635
## lag period: 0 3
## total df:  9
##
## BASIS FOR VAR:
## fun: ns
## knots: 18.27253 22.91653
## intercept: FALSE
## Boundary.knots: 8.551893 27.17635
##
## BASIS FOR LAG:
## fun: ns
```

```
## knots: 1
## intercept: TRUE
## Boundary.knots: 0 3
```

```
# DEFINE THE STRATA
london[, stratum:=factor(paste(MSOA11CD, year, month, sep=":"))]

# RUN THE MODEL
# NB: EXCLUDE EMPTY STRATA, OTHERWISE BIAS IN gnm WITH quasipoisson
london[,  keep:=sum(dtot)>0, by=stratum]
modfull <- gnm(dtot ~ cbtmean + spldoy:factor(year) + factor(dow),
  eliminate=stratum, data=london, family=quasipoisson, subset=keep)
```

Make predictions and plots

```
# PREDICT
temp <- seq(min(london$tmean),max(london$tmean),by=0.1)
cpfull <- crosspred(cbtmean, modfull, cen=16, at=temp)
# find minimum morbidity/mortality temperature
mmt <- temp[which.min(cpfull$allfit)]

cp_mmt <- crosspred(cbtmean, modfull, cen=mmt, at=temp)

# PLOT
col <- "darkgoldenrod3"
plot(cp_mmt, "overall", ylim=c(0.8,1.8), ylab="RR", col=col[1], lwd=1.5,
  xlab=expression(paste("Temperature ("*degree,"C)")),
  ci.arg=list(col=alpha(col[1], 0.2)))
```



Statistical analysis with space-time-stratified case crossover design

The space-time-stratified case crossover design doesn't require a complete time series, but an aggregated dataset is needed. Each combination of MSOA and date is an independent riskset.

Create case crossover data structure in June-August 2006 and 2013

2006 case crossover data structure

```
ds <- data.frame(case_date=as.Date(character()),
                 control_date1=as.Date(character()), control_date2=as.Date(character()),
                 control_date3=as.Date(character()), control_date4=as.Date(character()),
                 control_date5=as.Date(character()), control_date6=as.Date(character()),
                 control_date7=as.Date(character()), control_date8=as.Date(character()),
                 control_date9=as.Date(character()), stringsAsFactors = FALSE)
series <- data.frame(case_date=seq(as.Date('2006-06-01'), as.Date('2006-08-31'),'day'))

for(i in seq(nrow(series))){
  cat(i, '')
  ds[i, 1] <- series$case_date[i]
  ds[i, 2] <- series$case_date[i]-28
  ds[i, 3] <- series$case_date[i]-21
  ds[i, 4] <- series$case_date[i]-14
  ds[i, 5] <- series$case_date[i]-7
  ds[i, 6] <- series$case_date[i]
```

```
  ds[i, 7] <- series$case_date[i]+7
  ds[i, 8] <- series$case_date[i]+14
  ds[i, 9] <- series$case_date[i]+21
  ds[i, 10] <- series$case_date[i]+28
}

ds_2006 <- ds %>% gather('var','control_date', control_date1:control_date9) %>%
  filter(month(case_date)==month(control_date)) %>% select(-var) %>%
  mutate(case=case_when(case_date==control_date~1, TRUE~0))
```

2013 case crossover data structure

```
ds <- data.frame(case_date=as.Date(character()),
                 control_date1=as.Date(character()), control_date2=as.Date(character()),
                 control_date3=as.Date(character()), control_date4=as.Date(character()),
                 control_date5=as.Date(character()), control_date6=as.Date(character()),
                 control_date7=as.Date(character()), control_date8=as.Date(character()),
                 control_date9=as.Date(character()), stringsAsFactors = FALSE)
series <- data.frame(case_date=seq(as.Date('2013-06-01'), as.Date('2013-08-31'),'day'))

for(i in seq(nrow(series))){
  cat(i, '')
  ds[i, 1] <- series$case_date[i]
  ds[i, 2] <- series$case_date[i]-28
  ds[i, 3] <- series$case_date[i]-21
  ds[i, 4] <- series$case_date[i]-14
  ds[i, 5] <- series$case_date[i]-7
  ds[i, 6] <- series$case_date[i]
  ds[i, 7] <- series$case_date[i]+7
  ds[i, 8] <- series$case_date[i]+14
  ds[i, 9] <- series$case_date[i]+21
  ds[i, 10] <- series$case_date[i]+28
}

ds_2013 <- ds %>% gather('var','control_date', control_date1:control_date9) %>%
  filter(month(case_date)==month(control_date)) %>% select(-var) %>%
  mutate(case=case_when(case_date==control_date~1, TRUE~0))
```

Merge 2006 and 2013 case crossover data structure

```
ds <- rbind(ds_2006,ds_2013)
```

Expand the aggregated dataset and merge with environmental data. Lag structure is produced manually because the expanded dataset doesn't have a typical time series structure.

```
# separate the environmental data with outcome data for merging
# filter MSOA-days that have cases and phase out those without cases
london_cc <- london %>% select(MSOA11CD, date, dtot, year, month, day, doy, dow) %>%
  rename(case_date=date) %>% filter(dtot>0)
envdata <- london %>% select(MSOA11CD, date, tmean)

# generate lag structure for envdata
```

```r
envdata <- envdata %>% group_by(MSOA11CD) %>%
  arrange(date) %>%
  mutate(lag1=lag(tmean, 1), lag2=lag(tmean, 2),
         lag3=lag(tmean, 3)) %>%
  rename(control_date=date)

# expand mortality time series
dat <- left_join(london_cc, ds, by="case_date")

# merge with envdate
datfull <- left_join(dat, envdata, by= c("control_date","MSOA11CD"))
```

Set model parameters and fit model. Customize the crossbasis function consistent with CTS manually.

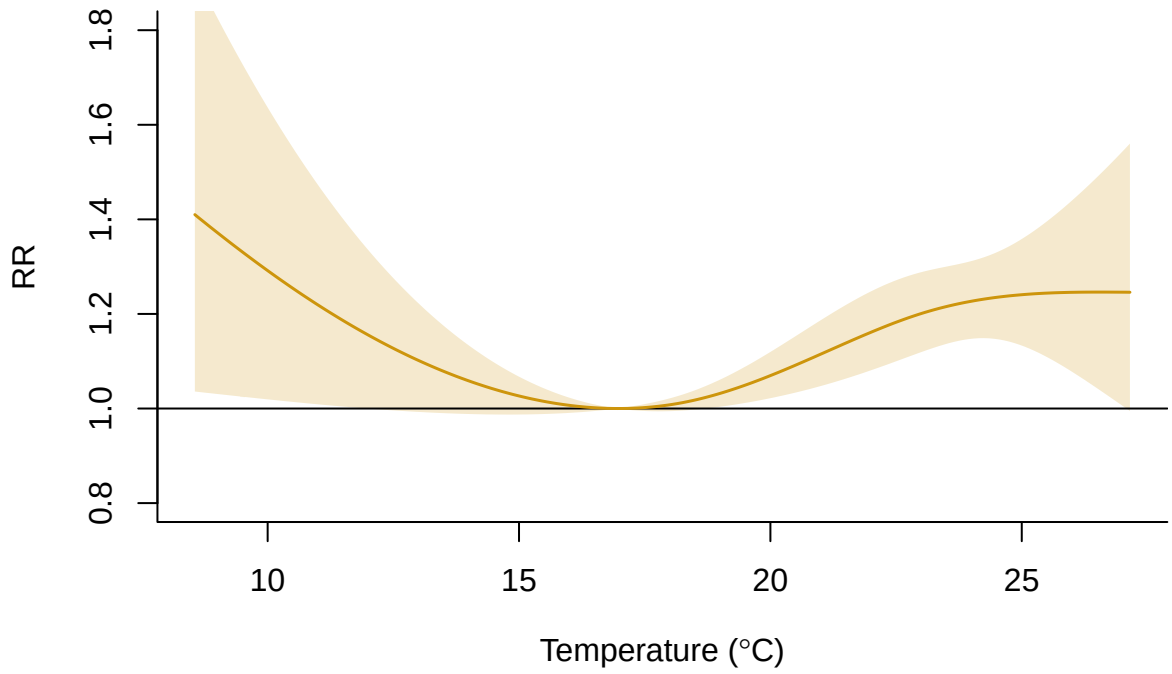
```r
# Note: the "datfull" full dataset has duplicated exposure data.
# The distribution of exposure from this dataset may not represent the real distribution
# of the exposure.

# DEFINE THE CROSS-BASIS FOR TEMPERATURE FROM THE EXPOSURE HISTORY MATRIX
argvar <- list(fun="ns", knots=quantile(datfull$tmean, c(50,90)/100, na.rm=T))
arglag <- list(fun="ns", knots=1)

# Include the temperature column as well as the lag columns.
cbtmean <- crossbasis(datfull[,11:14], lag=3, argvar=argvar, arglag=arglag)

summary(cbtmean)
```

```
## CROSSBASIS FUNCTIONS
## observations: 95101
## range: 8.551893 to 27.17635
## lag period: 0 3
## total df:  9
##
## BASIS FOR VAR:
## fun: ns
## knots: 18.28545 22.99516
## intercept: FALSE
## Boundary.knots: 8.551893 27.17635
##
## BASIS FOR LAG:
## fun: ns
## knots: 1
## intercept: TRUE
## Boundary.knots: 0 3
```

```r
# DEFINE THE STRATA
# Space-time-stratified case crossover design
datfull[, strata:=factor(paste(MSOA11CD, year, month, dow, sep=":"))]


# fit the conditional logistic model
mod_cc <- clogit(case ~ cbtmean + strata(strata), data=datfull, weights = dtot,
                          method = 'breslow') #put dtot in case weights
```

Make predictions and plots

```
# PREDICT
temp <- seq(min(london$tmean),max(london$tmean),by=0.1)
cpfull <- crosspred(cbtmean, mod_cc, cen=16, at=temp)
# find minimum morbidity/mortality temperature
mmt <- temp[which.min(cpfull$allfit)]

cp_mmt <- crosspred(cbtmean, mod_cc, cen=mmt, at=temp)

# PLOT
col <- "darkgoldenrod3"
plot(cp_mmt, "overall", ylim=c(0.8,1.8), ylab="RR", col=col[1], lwd=1.5,
  xlab=expression(paste("Temperature (*degree,"C)")),
  ci.arg=list(col=alpha(col[1], 0.2)))
```



Conclusion

Although we didn't choose a time-stratified CTS design, the exposure-response curves from small-are CTS and space-time-stratified case crossover design look very close. Both can be used in climate and health studies based on the discretion of the authors.

CTS has some advantages over case crossover design:

1. flexible structure to adjust for temporal and seasonal variation.

2. flexible lag configurations to examine the lag structure, no need to set the lag structure manually.

Case crossover design doesn't require a complete time series, while CTS does and excludes empty strata when running the model

Reference

1. Gasparrini A. A tutorial on the case time series design for small-area analysis. BMC Med Res Methodol. 2022;22(1):129. doi:10.1186/s12874-022-01612-x
2. Wu Y, Li S, Guo Y. Space-Time-Stratified Case-Crossover Design in Environmental Epidemiology Study. Health Data Science. 2021. doi:10.34133/2021/9870798