Impact of Sequence Characteristics on Posterior Changes

Author: Zhile Xu

UUN: s2500393

The code below explores the impact of sequence characteristics on the posterior changes of ETAS model parameters using synthetic earthquake catalogues. The analysis includes setting up the environment, generating synthetic data, fitting the ETAS model, and evaluating its performance under various scenarios.

Set up the environment

```
# Remove all objects from the current R environment to start fresh
rm(list = ls())
```

```
# Load the required libraries
library(ETAS.inlabru)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
set.seed(123)
```

```
# Set up parallel processing
num.cores <- 12
future::plan(future::multisession, workers = num.cores)
INLA::inla.setOption(num.threads = num.cores)
```

Set the true ETAS parameters

```
# set true ETAS parameters
true.param <- list(
  mu = 0.30106014, K = 0.13611399, alpha = 2.43945301,
  c = 0.07098607, p = 1.17838741
```

```
)
df.true.param <- data.frame(
  x = unlist(true.param),
  param = names(true.param)
)
# set magnitude distribution parameter
beta.p <- 2.353157
# set cutoff magnitude
M0 <- 2.5
# set starting time of the synthetic catalogue
T1 <- 0
# set end time of the synthetic catalogue
T2 <- 1000
```

Define a function to generate synthetic catalogues

```
# Define a function to generate synthetic catalogues
generate_synthetic_catalogue <- function(
    theta,
    beta.p,
    M0,
    T1,
    T2,
    Ht = NULL,
    idx = NULL) {
  synth.cat.list <- generate_temporal_ETAS_synthetic(
    theta = theta,
    beta.p = beta.p,
    M0 = M0,
    T1 = T1,
    T2 = T2,
    Ht = Ht
  )
  synth.cat.df <- do.call(rbind, synth.cat.list)
  synth.cat.df$idx <- as.integer(idx)
  return(synth.cat.df)
}
```

Generate synthetic catalogues

```
# Generate 10 synthetic catalogues
synth.cat.list <- lapply(1:10, function(i) {
  generate_synthetic_catalogue(
    theta = true.param,
    beta.p = beta.p,
    M0 = M0,
    T1 = T1,
    T2 = T2,
    idx = i
```

```
  )
})
# Sort each catalogue by occurrence time and add event identifier
synth.cat.list <- lapply(synth.cat.list, function(df) {
  df <- df[order(df$ts), ]
  df$idx.p <- seq_len(nrow(df))
  return(df)
})
```

```
# Bind the synthetic catalogues into a single data.frame
synth.cat.df <- bind_rows(synth.cat.list, .id = "idx")
synth.cat.df <- synth.cat.df %>% tibble()

# Add columns to the synthetic catalogue data to store the number of events
count_df <- synth.cat.df %>%
  group_by(idx) %>%
  summarise(
    N = n(),
    N.bkg = sum(gen == 1),
    N.after = sum(gen > 1)
  ) %>%
  arrange(idx)

# Add the count data to the synthetic catalogue data
synth.cat.df <- left_join(synth.cat.df, count_df, by = "idx")

# Add a column to the synthetic catalogue data to indicate the event type
synth.cat.df$dgen <- case_when(
  synth.cat.df$gen == 1 ~ "Background",
  synth.cat.df$gen > 1 ~ "Triggered",
  .default = "Other"
)
```

Plot the synthetic catalogues to visualise the distribution of events

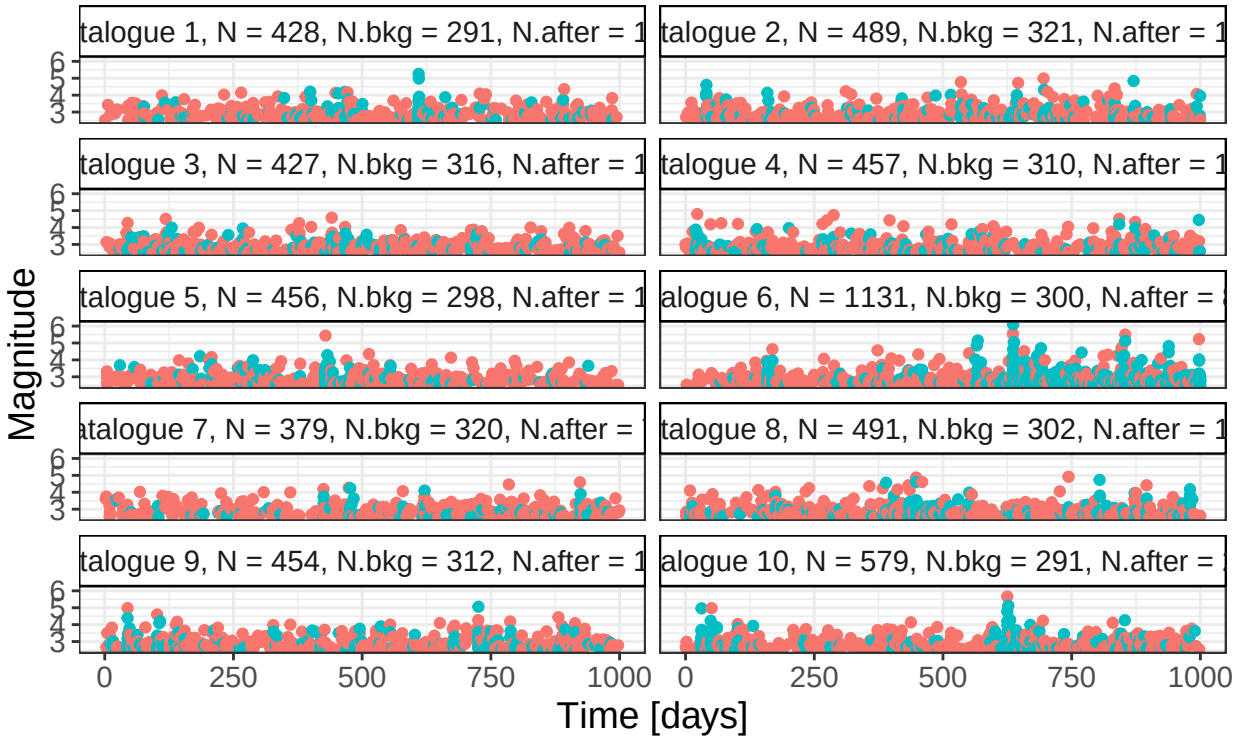
```
# Convert idx to integer
synth.cat.df$idx <- as.integer(synth.cat.df$idx)
# Convert idx to factor and sort the levels
synth.cat.df$idx <- factor(synth.cat.df$idx,
  levels = sort(unique(synth.cat.df$idx))
)
```

```
library(dplyr)
plot.synth <- synth.cat.df %>%
  ggplot(aes(x = ts, y = magnitudes)) +
  geom_point(aes(color = factor(dgen))) +
  facet_wrap(~idx,
    scales = "fixed", ncol = 2,
    labeller = as_labeller(function(x) {
      paste0(
        "Catalogue ", x, ", N = ", unique(synth.cat.df$N[synth.cat.df$idx == x]),
```

```r
        ", N.bkg = ", unique(synth.cat.df$N.bkg[synth.cat.df$idx == x]),
        ", N.after = ",
        unique(synth.cat.df$N.after[synth.cat.df$idx == x])
      )
    })
  ) +
  theme_bw() +
  theme(
    text = element_text(size = 14),
    legend.position = "bottom",
    strip.background = element_rect(fill = "white", color = "black"),
  ) +
  labs(x = "Time [days]", y = "Magnitude") +
  scale_color_discrete(name = "Event Type")

plot.synth
```

```
## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```




```
ggsave("q2-synth-cat-plot.png",
  plot = plot.synth,
  height = 8.27, width = 11.69, units = "in", dpi = 300
)
```

```
## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length
## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length
```

```
## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length

## Warning in `==.default`(synth.cat.df$idx, x): longer object length is not a
## multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

Fit the ETAS model to the synthetic catalogues

```
# set copula transformations list
link.f <- list(
  mu = \(x) gamma_t(x, 0.3, 0.6),
  K = \(x) unif_t(x, 0, 10),
  alpha = \(x) unif_t(x, 0, 10),
  c_ = \(x) unif_t(x, 0, 10),
  p = \(x) unif_t(x, 1, 10)
)

# set inverse copula transformations list
inv.link.f <- list(
  mu = \(x) inv_gamma_t(x, 0.3, 0.6),
  K = \(x) inv_unif_t(x, 0, 10),
  alpha = \(x) inv_unif_t(x, 0, 10),
  c_ = \(x) inv_unif_t(x, 0, 10),
  p = \(x) inv_unif_t(x, 1, 10)
)

# set up list of initial values
th.init <- list(
  th.mu = inv.link.f$mu(0.5),
  th.K = inv.link.f$K(0.1),
  th.alpha = inv.link.f$alpha(1),
  th.c = inv.link.f$c_(0.1),
  th.p = inv.link.f$p(1.1)
)

# set up list of bru options
bru.opt.list <- list(
  bru_verbose = 0, # type of visual output
  bru_max_iter = 70, # maximum number of iterations
```

```
  # bru_method = list(max_step = 0.5),
  bru_initial = th.init
) # parameters initial values
```

```
# create a list of model fitting results
fit.list <- lapply(synth.cat.list, function(synth.cat.df) {
  # fit the model
  fit <- Temporal.ETAS(
    total.data = synth.cat.df,
    M0 = M0,
    T1 = T1,
    T2 = T2,
    link.functions = link.f,
    coef.t. = 1,
    delta.t. = 0.1,
    N.max. = 5,
    bru.opt = bru.opt.list
  )
  print("finish...")
  return(fit)
})
```

```
## Start creating grid...
## Finished creating grid, time  0.454335
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  0.5465071
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  0.461823
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  0.5129039
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  0.5042939
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  1.387863
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  0.4083419
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  0.54673
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  0.4826391
## [1] "finish..."
## Start creating grid...
## Finished creating grid, time  0.6589141
## [1] "finish..."
```

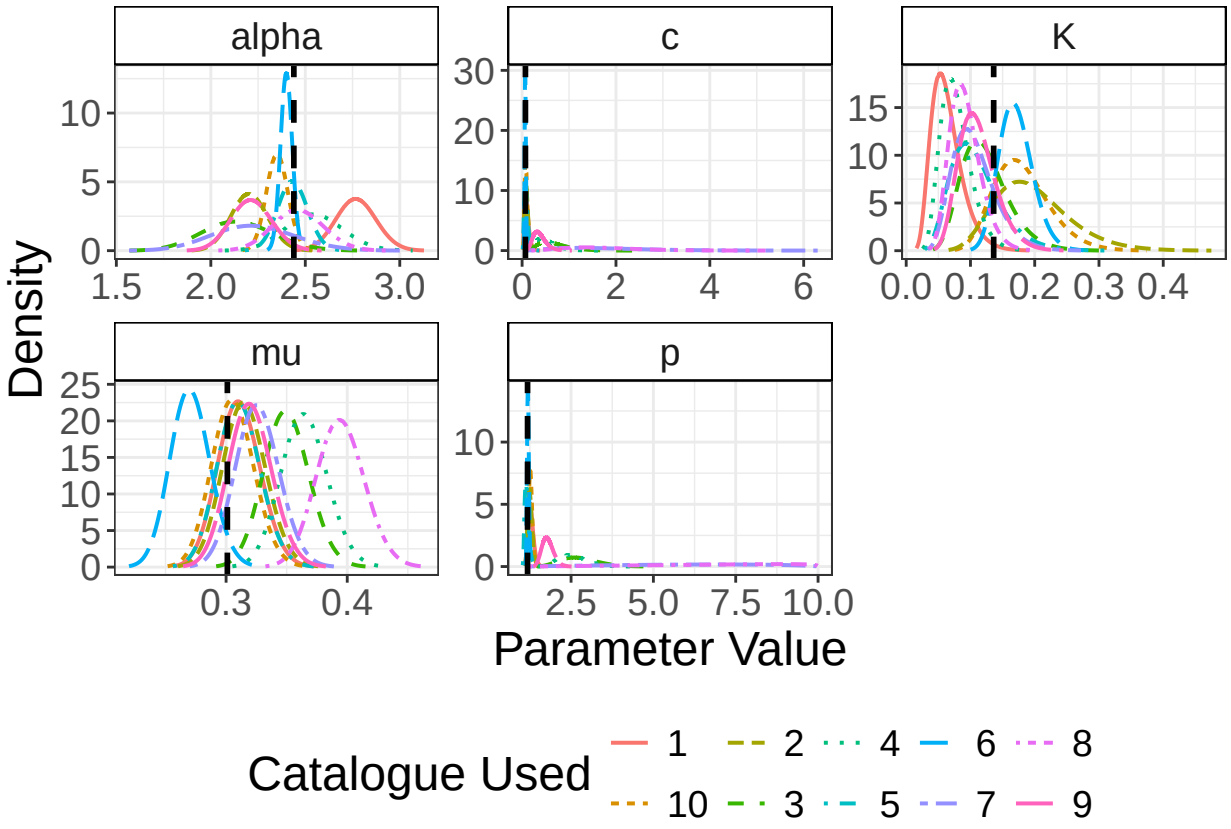
Plot the ETAS model posterior distributions

```
# create a list of model fitting results
input_lists <- lapply(1:length(fit.list), function(i) {
  list(
    model.fit = fit.list[[i]],
    link.functions = link.f
  )
})

# retrieve marginal posterior distributions and set model identifier
post_lists <- lapply(1:length(fit.list), function(i) {
  post.list <- get_posterior_param(input.list = input_lists[[i]])
  post.list$post.df$cat.used <- as.character(i)
  return(post.list)
})

# bind marginal posterior data.frames
bind.post.df <- do.call(rbind, lapply(post_lists, function(x) x$post.df))
```

```
# Plot the marginal posterior distributions of the ETAS parameters
plot.mp <- ggplot(bind.post.df, aes(
  x = x, y = y,
  color = cat.used, linetype = cat.used
)) +
  geom_line(size = 0.7) +
  facet_wrap(~param, scales = "free", ncol = 3) +
  xlab("Parameter Value") +
  ylab("Density") +
  geom_vline(
    data = df.true.param,
    aes(xintercept = x), linetype = 2, color = "black", size = 1
  ) +
  theme_bw() +
  theme(
    text = element_text(size = 18),
    legend.position = "bottom",
    strip.background = element_rect(fill = "white", color = "black"),
  ) +
  labs(
    color = "Catalogue Used",
    linetype = "Catalogue Used"
  )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
plot.mp
```




```
ggsave("q2-posterior-distributions.png",
  plot = plot.mp,
  height = 8.27, width = 11.69, units = "in", dpi = 300
)
```