Simulated example

Jouni Helske and Santtu Tikka

2024-02-01

Load some packages and create the data:

```
library(dynamite)
library(dplyr)
library(ggplot2)
library(RColorBrewer)
```

fix number of individuals and number of time points:

```
# number of individuals
m <- 500
# number of time points before intervention
n1 <- 80
# number of time points after the first intervention
# (inluding the first time point of intervention)
n2 <- 20
# total number of time points
n <- n1 + n2
```

Function for simulating the data:

```
simulate_data <- function(scenario) {

  sigma_y <- sigma_x <- 0.4
  beta_yx <- 0.4
  beta_yy <- 0.6
  beta_xx <- 0.9
  beta_xy <- -0.1
  beta_xz <- 0.4
  beta_yz <- 1
  if (scenario == 1) {
    z <- rep(c(0, 1, 0), times = c(n1, 1, n2 - 1))
  } else {
    z <- rep(0:1, times = c(n1, n2))
  }
  # with intervention
  x <- y <- matrix(0, m, n)
  # no intervention
  x_ <- y_ <- matrix(0, m, n)
  # means
  mean_y <- mean_y_ <- matrix(0, m, n)
  x[, 1] <- x_[, 1] <- rnorm(m)
  y[, 1] <- y_[, 1] <- rnorm(m)
  for(i in 2:n1) {
    mean_y[, i]  <- beta_yy * y[, i - 1] + beta_yx * x[, i - 1]
```

```
    mean_y_[, i] <- mean_y[, i]
    e_x <- rnorm(m, 0, sigma_x)
    e_y <- rnorm(m, 0, sigma_y)
    x[, i] <- beta_xy * y[, i - 1] + beta_xx * x[, i - 1] + e_x
    y[, i] <- mean_y[, i] + e_y
    x_[, i] <- x[, i]
    y_[, i] <- y[, i]
  }
  for(i in (n1 + 1):n) {
    mean_y[, i]  <- beta_yz * z[i] + beta_yy * y[, i - 1] + beta_yx * x[, i - 1]
    mean_y_[, i] <- beta_yy * y_[, i - 1] + beta_yx * x_[, i - 1]
    e_x <- rnorm(m, 0, sigma_x)
    e_y <- rnorm(m, 0, sigma_y)
    x[, i] <- beta_xz * z[i] + beta_xy * y[, i - 1] + beta_xx * x[, i - 1] + e_x
    y[, i] <- mean_y[, i] + e_y
    x_[, i] <- beta_xy * y_[, i - 1] + beta_xx * x_[, i - 1] + e_x
    y_[, i] <- mean_y_[, i] + e_y
  }
  data.frame(
    y = c(t(y)),
    x = c(t(x)),
    y_ = c(t(y_)),
    x_ = c(t(x_)),
    mean_y = c(t(mean_y)),
    mean_y_ = c(t(mean_y_)),
    z = z,
    time = 1:n,
    id = rep(factor(1:m), each = n))
}
```

Scenario 1

Create data:

```
set.seed(808)
# Data for the atomic case
d_1 <- simulate_data(scenario = 1)
true_effect_1 <- d_1 |>
  filter(time > n1) |>
  group_by(time) |>
  summarise(mean = mean(mean_y - mean_y_))
```

Estimate the model with dynamite:

```
# Estimate the model
fit_1 <- dynamite(
  obs(y ~ z, family = "gaussian") + obs(x ~ z, family = "gaussian") + lags(),
  data = d_1,
  time = "time",
  group = "id",
  chains = 4, cores = 4, refresh = 0)
saveRDS(fit_1, file = "fit_simulated_scenario1.rds")
```

Check MCMC diagnostics:

```r
mcmc_diagnostics(fit_1)
```

```
## NUTS sampler diagnostics:
##
## No divergences, saturated max treedepths or low E-BFMIs.
##
## Smallest bulk-ESS values:
##
## beta_x_z      3328
## beta_y_z      3647
## beta_y_x_lag1 5045
##
## Smallest tail-ESS values:
##
## beta_x_z 2441
## alpha_x  2912
## sigma_y  3060
##
## Largest Rhat values:
##
## beta_y_x_lag1 1
## beta_x_y_lag1 1
## alpha_x       1
```

Parameter estimates:

```r
as_draws(fit_1) |>
  posterior::summarise_draws(
    "mean",
    "sd",
    ~quantile(.x, probs = c(0.025, 0.975)),
     "rhat", "ess_bulk", "ess_tail")
```

```
## # A tibble: 10 x 8
##    variable           mean      sd   `2.5%`   `97.5%`  rhat ess_bulk ess_tail
##    <chr>             <dbl>   <dbl>    <dbl>     <dbl> <dbl>    <dbl>    <dbl>
##  1 alpha_x       -0.00374  0.00180 -0.00735 -0.000171  1.00    8121.    2912.
##  2 alpha_y       -0.000284 0.00177 -0.00370  0.00333   1.00    6569.    3210.
##  3 beta_x_x_lag1  0.899    0.00280  0.894    0.905     1.00    5897.    3281.
##  4 beta_x_y_lag1 -0.0975   0.00265 -0.103   -0.0922    1.00    5591.    3188.
##  5 beta_x_z       0.386    0.0180   0.351    0.421     1.00    3328.    2441.
##  6 beta_y_x_lag1  0.394    0.00277  0.388    0.399     1.00    5045.    3347.
##  7 beta_y_y_lag1  0.604    0.00259  0.599    0.609     1.00    5249.    3160.
##  8 beta_y_z       0.987    0.0176   0.951    1.02      1.00    3647.    3217.
##  9 sigma_x        0.402    0.00130  0.400    0.405     1.00    6507.    3153.
## 10 sigma_y        0.402    0.00130  0.400    0.405     1.00    6930.    3060.
```

Estimate the causal effects:

```r
newdata <- d_1 |>
  mutate(
    y = ifelse(time > n1, NA, y),
    x = ifelse(time > n1, NA, x)
  ) |>
  filter(time >= n1)
```

```r
intervention_correct <- predict(
  fit_1, newdata = newdata, type = "mean", funs = list(y = list(mean = mean))
)$simulated

newdata <- d_1 |>
  mutate(
    y = ifelse(time > n1, NA, y),
    x = ifelse(time > n1, NA, x),
    z = 0
  ) |>
  filter(time >= n1)

no_intervention_correct <- predict(
  fit_1, newdata = newdata, type = "mean", funs = list(y = list(mean = mean))
)$simulated

newdata <- d_1 |>
  mutate(
    y = ifelse(time > n1, NA, y)
  ) |>
  filter(time >= n1)

intervention_incorrect <- predict(
  fit_1, newdata = newdata, type = "mean", funs = list(y = list(mean = mean))
)$simulated

newdata <- d_1 |>
  mutate(
    y = ifelse(time > n1, NA, y),
    z = 0
  ) |>
  filter(time >= n1)

no_intervention_incorrect <- predict(
  fit_1, newdata = newdata, type = "mean", funs = list(y = list(mean = mean))
)$simulated

results_1 <- bind_rows(
  correct = bind_rows(
    yes = intervention_correct,
    no = no_intervention_correct,
    .id = "intervention"
  ),
  incorrect = bind_rows(
    yes = intervention_incorrect,
    no = no_intervention_incorrect,
    .id = "intervention"
  ),
  .id = "Method"
) |>
  filter(time > n1) |>
  group_by(Method, time, .draw) |>
  summarise(
```

```
    difference = mean_y[intervention == "yes"] - mean_y[intervention == "no"]
  ) |>
  group_by(Method, time) |>
  summarise(
    mean = mean(difference),
    q2.5 = quantile(difference, 0.025),
    q97.5 = quantile(difference, 0.975),
    q10 = quantile(difference, 0.1),
    q90 = quantile(difference, 0.9),
  )
saveRDS(results_1, file = "results_scenario1.rds")
```

Scenario 2

Create data:

```
set.seed(808)
# Data for the recurring case
d_2 <- simulate_data(scenario = 2)
true_effect_2 <- d_2 |>
  filter(time > n1) |>
  group_by(time) |>
  summarise(mean = mean(mean_y - mean_y_))
```

Estimate the model with dynamite:

```
# Estimate the model
fit_2 <- dynamite(
  obs(y ~ z, family = "gaussian") + obs(x ~ z, family = "gaussian") + lags(),
  data = d_2,
  time = "time",
  group = "id",
  chains = 4, cores = 4, refresh = 0)
saveRDS(fit_2, file = "fit_simulated_scenario2.rds")
```

Check MCMC diagnostics:

```
mcmc_diagnostics(fit_2)
```

```
## NUTS sampler diagnostics:
##
## No divergences, saturated max treedepths or low E-BFMIs.
##
## Smallest bulk-ESS values:
##
## beta_y_z      3027
## beta_y_y_lag1 3360
## beta_x_z      3474
##
## Smallest tail-ESS values:
##
## beta_x_x_lag1 2563
## beta_x_z      2582
## alpha_y       2686
##
## Largest Rhat values:
```

```
##
## sigma_y       1
## beta_y_x_lag1 1
## beta_x_z      1
```

Parameter estimates:

```
as_draws(fit_2) |>
  posterior::summarise_draws(
    "mean",
    "sd",
    ~quantile(.x, probs = c(0.025, 0.975)),
     "rhat", "ess_bulk", "ess_tail")
```

```
## # A tibble: 10 x 8
##    variable           mean      sd   `2.5%`   `97.5%`  rhat ess_bulk ess_tail
##    <chr>             <dbl>   <dbl>    <dbl>     <dbl> <dbl>    <dbl>    <dbl>
##  1 alpha_x       -0.00467  0.00199 -0.00865 -0.000818  1.00    5939.    3294.
##  2 alpha_y       -0.000125 0.00201 -0.00409  0.00380   1.00    5341.    2686.
##  3 beta_x_x_lag1  0.899    0.00280  0.894    0.905     1.00    4557.    2563.
##  4 beta_x_y_lag1 -0.0973   0.00242 -0.102   -0.0926    1.00    3480.    2716.
##  5 beta_x_z       0.397    0.00745  0.382    0.412     1.00    3474.    2582.
##  6 beta_y_x_lag1  0.394    0.00277  0.389    0.399     1.00    4776.    3055.
##  7 beta_y_y_lag1  0.602    0.00236  0.598    0.607     1.00    3360.    3185.
##  8 beta_y_z       0.996    0.00745  0.982    1.01      1.00    3027.    2973.
##  9 sigma_x        0.402    0.00126  0.400    0.405     1.00    5929.    2849.
## 10 sigma_y        0.402    0.00125  0.400    0.405     1.00    5725.    2956.
```

Estimate the causal effects:

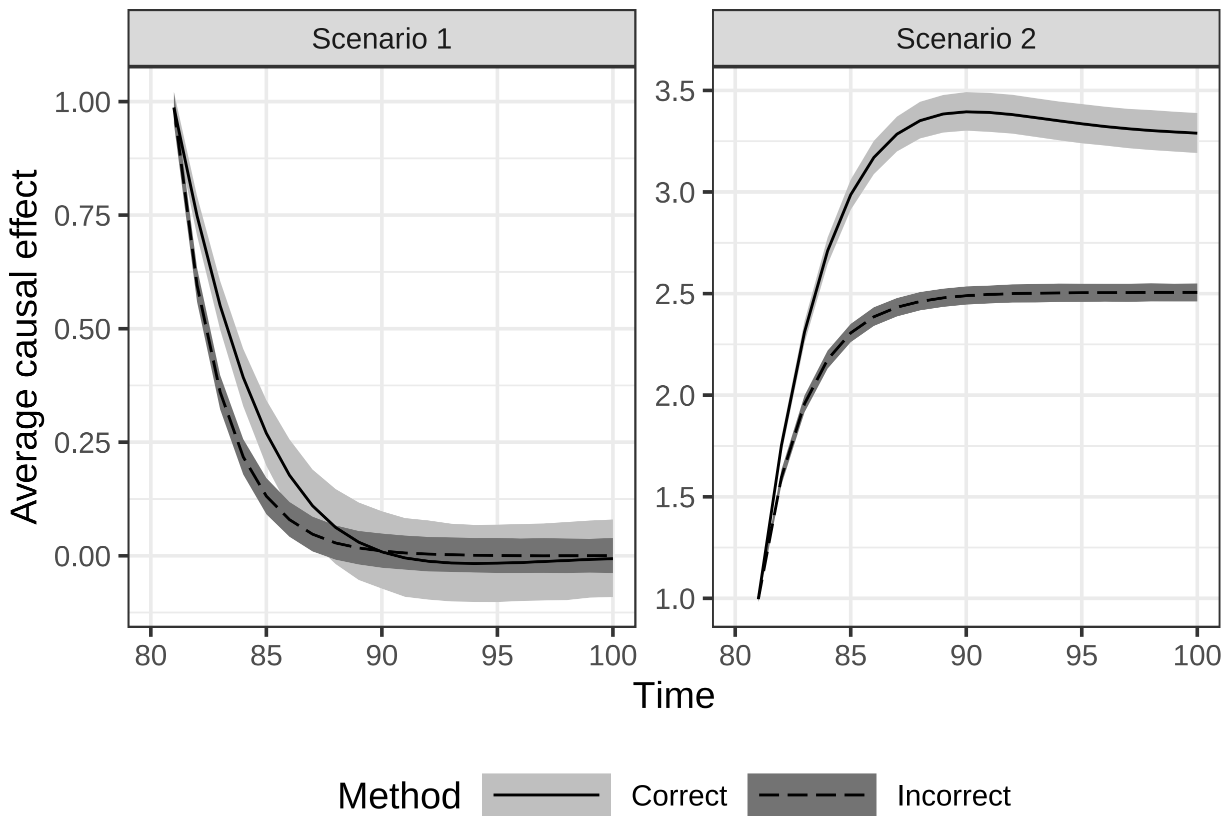
```
newdata <- d_2 |>
  mutate(
    y = ifelse(time > n1, NA, y),
    x = ifelse(time > n1, NA, x)
  ) |>
  filter(time >= n1)

intervention_correct <- predict(
  fit_2, newdata = newdata, type = "mean", funs = list(y = list(mean = mean))
)$simulated

newdata <- d_2 |>
  mutate(
    y = ifelse(time > n1, NA, y),
    x = ifelse(time > n1, NA, x),
    z = 0
  )  |>
  filter(time >= n1)

no_intervention_correct <- predict(
  fit_2, newdata = newdata, type = "mean", funs = list(y = list(mean = mean))
)$simulated

newdata <- d_2 |>
  mutate(
    y = ifelse(time > n1, NA, y)
```

```r
  ) |>
  filter(time >= n1)

intervention_incorrect <- predict(
  fit_2, newdata = newdata, type = "mean", funs = list(y = list(mean = mean))
)$simulated

newdata <- d_2 |>
  mutate(
    y = ifelse(time > n1, NA, y),
    z = 0
  ) |>
  filter(time >= n1)

no_intervention_incorrect <- predict(
  fit_2, newdata = newdata, type = "mean", funs = list(y = list(mean = mean))
)$simulated

results_2 <- bind_rows(
  correct = bind_rows(
    yes = intervention_correct,
    no = no_intervention_correct,
    .id = "intervention"
  ),
  incorrect = bind_rows(
    yes = intervention_incorrect,
    no = no_intervention_incorrect,
    .id = "intervention"
  ),
  .id = "Method"
) |>
  filter(time > n1) |>
  group_by(Method, time, .draw) |>
  summarise(
    difference = mean_y[intervention == "yes"] - mean_y[intervention == "no"]
  ) |>
  group_by(Method, time) |>
  summarise(
    mean = mean(difference),
    q2.5 = quantile(difference, 0.025),
    q97.5 = quantile(difference, 0.975),
    q10 = quantile(difference, 0.1),
    q90 = quantile(difference, 0.9),
  )
saveRDS(results_2, file = "results_scenario2.rds")
```

Figures for the paper

```r
results_1 <- readRDS("results_scenario1.rds")
results_2 <- readRDS("results_scenario2.rds")
p <- bind_rows(
  `Scenario 1` = results_1,
  `Scenario 2` = results_2,
```

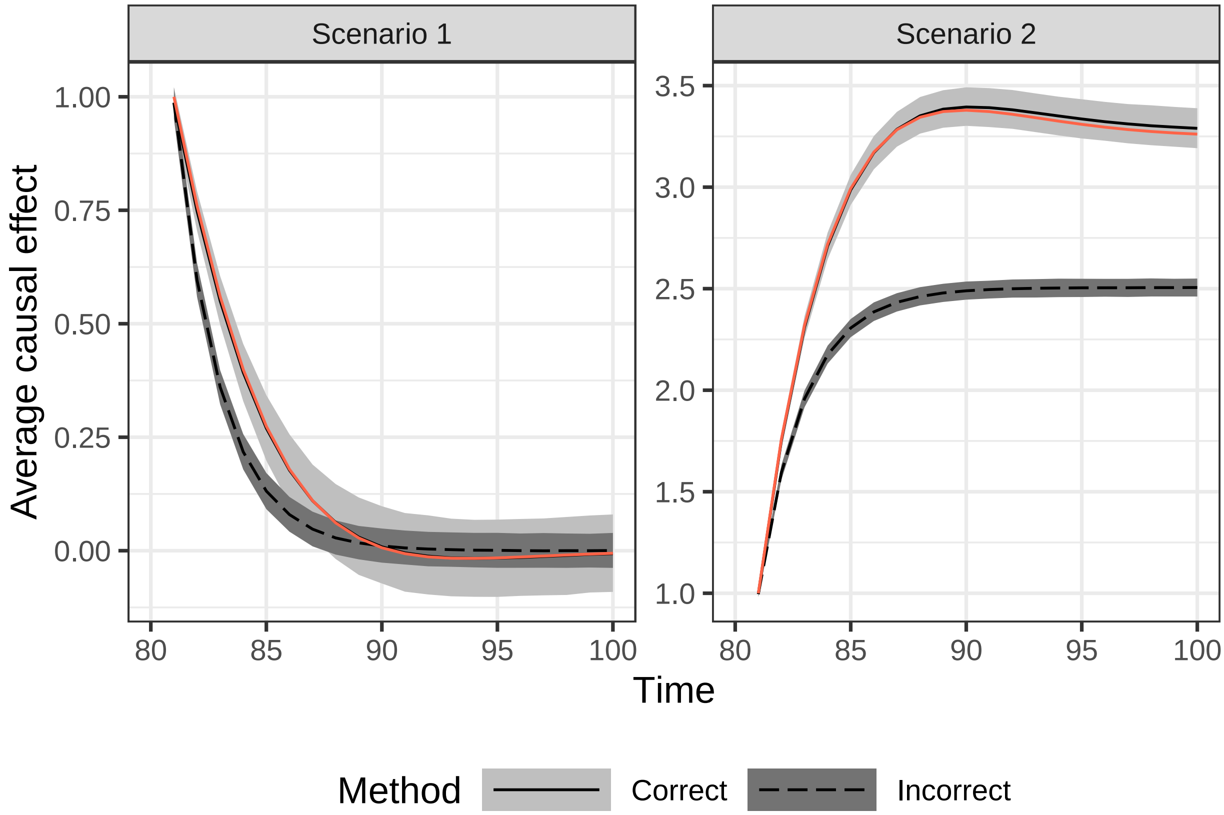
```r
  .id = "Intervention"
) |>
  ggplot(aes(time, mean)) +
  geom_ribbon(aes(ymin = q2.5, ymax = q97.5, fill = Method)) +
  geom_line(aes(linetype = Method)) +
  scale_fill_grey(
    start = 0.75,
    end = 0.45,
    labels = c("Correct", "Incorrect")
  ) +
  scale_linetype_manual(
    values = c("solid", "longdash"),
    labels = c("Correct", "Incorrect")
  ) +
  ylab("Average causal effect") +
  xlab("Time") + xlim(c(80, 100)) +
  theme_bw(base_size = 14) +
  theme(
    legend.position = "bottom",
    panel.grid.minor.x = element_blank(),
    legend.key.width = unit(1.75, "cm")
  ) +
  facet_wrap(~ Intervention, scales = "free")
p
```



```r
ggsave(p, file = "../ex1_results.png", width = 7, height = 4)
```

Same figure where the true effects are shown in red:

```r
p <- bind_rows(
  `Scenario 1` = results_1,
  `Scenario 2` = results_2,
  .id = "Intervention"
) |>
  ggplot(aes(time, mean)) +
  geom_ribbon(aes(ymin = q2.5, ymax = q97.5, fill = Method)) +
  geom_line(aes(linetype = Method)) +
  scale_fill_grey(
    start = 0.75,
    end = 0.45,
    labels = c("Correct", "Incorrect")
  ) +
  geom_line(data = cbind(Intervention = "Scenario 1", true_effect_1), colour = "tomato") +
    geom_line(data = cbind(Intervention = "Scenario 2", true_effect_2), colour = "tomato") +
  scale_linetype_manual(
    values = c("solid", "longdash"),
    labels = c("Correct", "Incorrect")
  ) +
  ylab("Average causal effect") +
  xlab("Time") + xlim(c(n1, n)) +
  theme_bw(base_size = 14) +
  theme(
    legend.position = "bottom",
    panel.grid.minor.x = element_blank(),
    legend.key.width = unit(1.75, "cm")
  ) +
  facet_wrap(~ Intervention, scales = "free")
p
```




```r
ggsave(p, file = "../ex1_results_with_truth.png", width = 7, height = 4)
```