simple example simulation

Siyue Yang

19/02/2021

```r
library(matrixStats)
library(kableExtra)


cal_bias_var <- function(x, y, label.size) {
  x.lab <- x[1: label.size]
  y.lab <- y[1: label.size]

  mod <- lm(y.lab~x.lab)

  sl <- mean(y.lab)
  sl.bias <- sl - mean(y)
  sl.sd <- sd(y.lab)

  x.unlab <- x[(label.size+1):length(y)]

  # m estimator
  mx <- summary(mod)$coef[1,1] + summary(mod)$coef[2,1] * x.unlab
  ssl <- mean(mx)
  ssl.bias <- ssl - mean(y)
  ssl.sd <- sd(mx)

  sl.mse <- sl.sd^2 + sl.bias^2
  ssl.mse <- ssl.sd^2 + sl.bias^2

  return(c(sl, sl.bias, sl.sd, sl.mse, ssl, ssl.bias, ssl.sd, ssl.ase = 0.5, ssl.mse))
}
```

```r
run_sim <- function(N, n_lab, coeff, nsim = 5000, eps_sd = 0.5) {
  linear <- c()
  nonlinear_1 <- c()
  nonlinear_2 <- c()

  for (nn in c(1:nsim)) {
    set.seed(1234 + nn)
    x <- rnorm(N, 0, 1)
    eps <- rnorm(N, mean=0, sd=0.5)

    y = coeff*x + eps
    linear <- rbind(linear, cal_bias_var(x, y, n_lab))

    w <- coeff*x + coeff*x^2 + eps
    nonlinear_1 <- rbind(nonlinear_1, cal_bias_var(x, w, n_lab))

    z <- coeff*x + coeff*x^2 + coeff*x^3 + eps
    nonlinear_2 <- rbind(nonlinear_2, cal_bias_var(x, z, n_lab))
  }

  return(list(linear = linear, nonlinear_1 = nonlinear_1, nonlinear_2 = nonlinear_2))
}
```

```
# Setting 1: compare n = 100, 200, 400
lab_100_beta_0.5 <- run_sim(10000, 100, 0.5)
lab_200_beta_0.5 <- run_sim(10000, 200, 0.5)
lab_400_beta_0.5 <- run_sim(10000, 400, 0.5)
```

```
table1 <- rbind(colMeans(lab_100_beta_0.5$linear), colMeans(lab_200_beta_0.5$linear), colMeans(lab_400_
rownames(table1) <- c(100, 200, 400)
colnames(table1) <- c("SL", "bias", "ESE", "MSE", "SSL", "Bias", "ESE", "ASE", "MSE")

Efficiency <- table1[, 4] / table1[, 9]
table1 <- cbind(table1, Efficiency)
table1 <- round(table1, 4)


kableExtra::kbl(table1, booktabs = T, caption="Efficiencies with respect to empirical mean square error
  add_header_above(c(" ", "Supervised learning estimator"=4, "Semi-supervised estimator"=5, " ")) %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

Table 1: Efficiencies with respect to empirical mean square error for m(x) = 0.5x

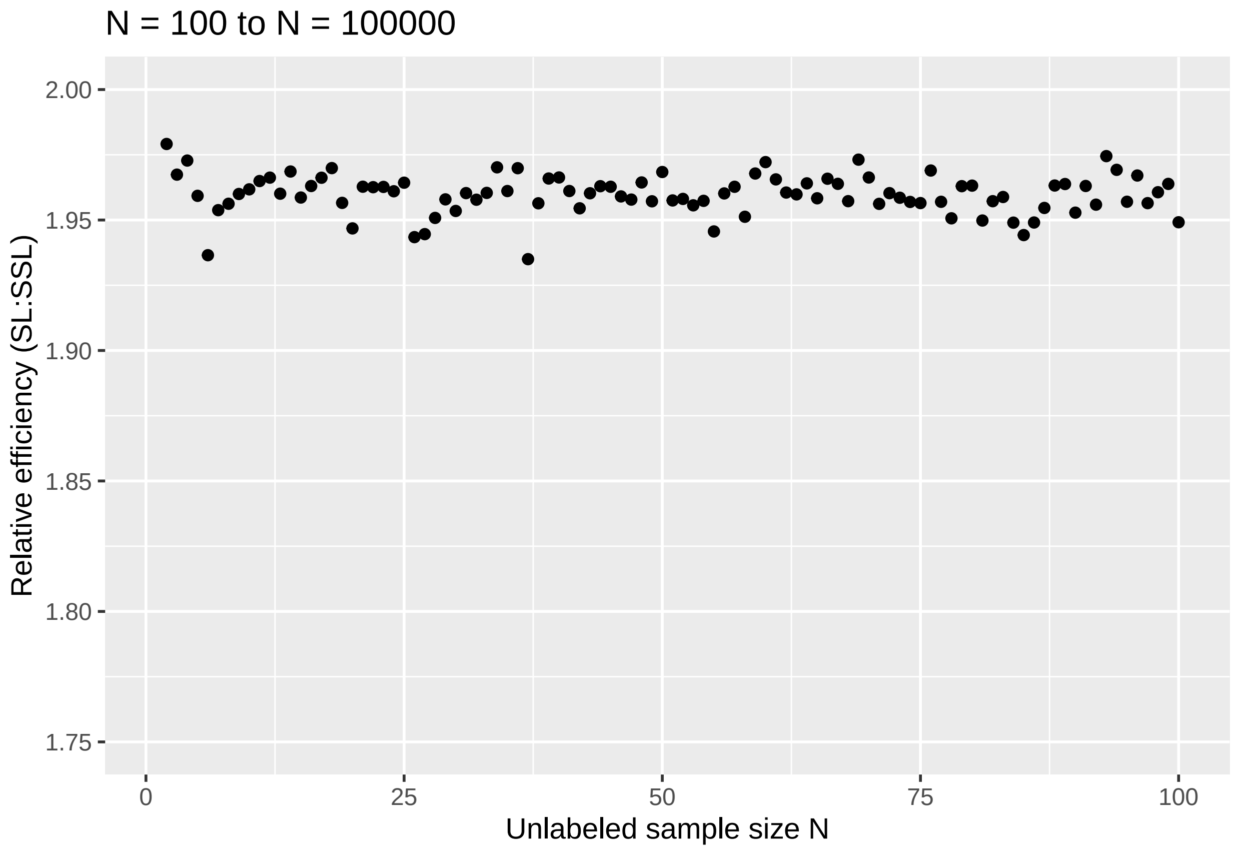
	Supervised learning estimator				Semi-supervised estimator					
	SL	bias	ESE	MSE	SSL	Bias	ESE	ASE	MSE	Efficiency
100	-0.0011	-0.0012	0.7052	0.5048	2e-04	2e-04	0.5002	0.5	0.2578	1.9580
200	-0.0004	-0.0004	0.7062	0.5025	6e-04	5e-04	0.5003	0.5	0.2542	1.9770
400	-0.0001	-0.0002	0.7064	0.5008	2e-04	1e-04	0.4996	0.5	0.2515	1.9912

```
# Setting 2: N=10000 down to n = 100
r <- c()
for (i in seq(100, 10000, 100)) {
  r <- rbind(r, colMeans(run_sim(i, 100, 0.5, nsim = 1000)$linear))
}
```

```
library(ggplot2)
Efficiency <- r[, 4] / r[, 9]
Efficiency[1] <- 1

data.frame(index = c(1:100), eff = Efficiency) %>%
  ggplot() + geom_point(aes(index, eff)) + ylim(c(1.75, 2)) +
  ylab("Relative efficiency (SL:SSL)") + xlab("Unlabeled sample size N") + ggtitle("N = 100 to N = 1000
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



```r
# library(ggplot2)
# # sa <- stack(as.data.frame(r[, c(7, 3)]))
# # sa['label'] <- seq(100, 10000, 100)
#
# data.frame(r[, c(7,3)])
# sa %>% ggplot(aes(x=factor(label), y=values)) +
#     geom_point(size = 0.7) + theme_bw() +
#   theme(axis.text.x = element_text(size = 5, angle = 90, vjust = 0.5, hjust=1)) +
#   ylab("Relative efficiency (SL:SSL)") + xlab("labeled size n")
```

```r
# Setting 3: compare m(x) = 0.5 and m(x) = 2
lab_400_beta_2 <- run_sim(10000, 400, 2)
```

```r
table3 <- rbind(colMeans(lab_400_beta_0.5$linear), colMeans(lab_400_beta_2$linear))
rownames(table3) <- c("m(x) = 0.5x", "m(x) = 2x")

colnames(table3) <- c("SL", "bias", "ESE", "MSE", "SSL", "Bias", "ESE", "ASE", "MSE")

Efficiency <- table3[, 4] / table3[, 9]
table3 <- cbind(table3, Efficiency)
table3 <- round(table3, 4)

kableExtra::kbl(table3, booktabs = T, caption="Compare m = 0.5x and m(x) = 2x") %>%
  add_header_above(c(" ", "Supervised learning estimator"=4, "Semi-supervised estimator"=5, " ")) %>%
  kable_styling(latex_options = c("scale_down", "hold_position"))
```

Table 2: Compare m = 0.5x and m(x) = 2x

	Supervised learning estimator				Semi-supervised estimator					
	SL	bias	ESE	MSE	SSL	Bias	ESE	ASE	MSE	Efficiency
m(x) = 0.5x	-1e-04	-2e-04	0.7064	0.5008	2e-04	1e-04	0.4996	0.5	0.2515	1.9912
m(x) = 2x	-8e-04	-1e-03	2.0606	4.2619	4e-04	1e-04	1.9997	0.5	4.0100	1.0628