Log likelihood accuracy by number of Monte Carlo samples

John Ensley

January 22, 2018

```
library(gpcovr)
set.seed(1234)
```

Parameter settings

```
n <- 400          # number of observations from GP
nu <- 1.5         # Matern parameter (smoothness)
alpha <- 0.25     # Matern parameter (inverse range)
sigma <- 1        # Matern parameter (variance)
tau <- 0.01       # Nugget effect
nsamp <- c(100, 500, 1000, 5000, 10000, 50000, 100000)
                  # number of MC samples to take when evaluating log likelihood
M <- 100          # number of repetitions at each of nsamp
```

Data generation

```
# generate the observation locations.
# n observed pts, 50x50 prediction grid (not used here)
locations <- create_locations(n, 50)
# uncomment to visualize locations:
# plot(locations)

# simulate the Gaussian process
gp <- simulate_gp(locations, 'matern', c(nu, 1/alpha, sigma, tau))
# uncomment to visualize GP:
# plot(gp)

# pull out vector of observations
observations <- gp$Y[gp$locs$type == 'obs']
# pull out n x n distance matrix
dist_mat <- gp$dist_obs
```

Exact log likelihood calculation

```
( exact <- normal_ll_exact(observations, dist_mat, nu, alpha, sigma, tau) )
```

```
## [1] 354.8083
```

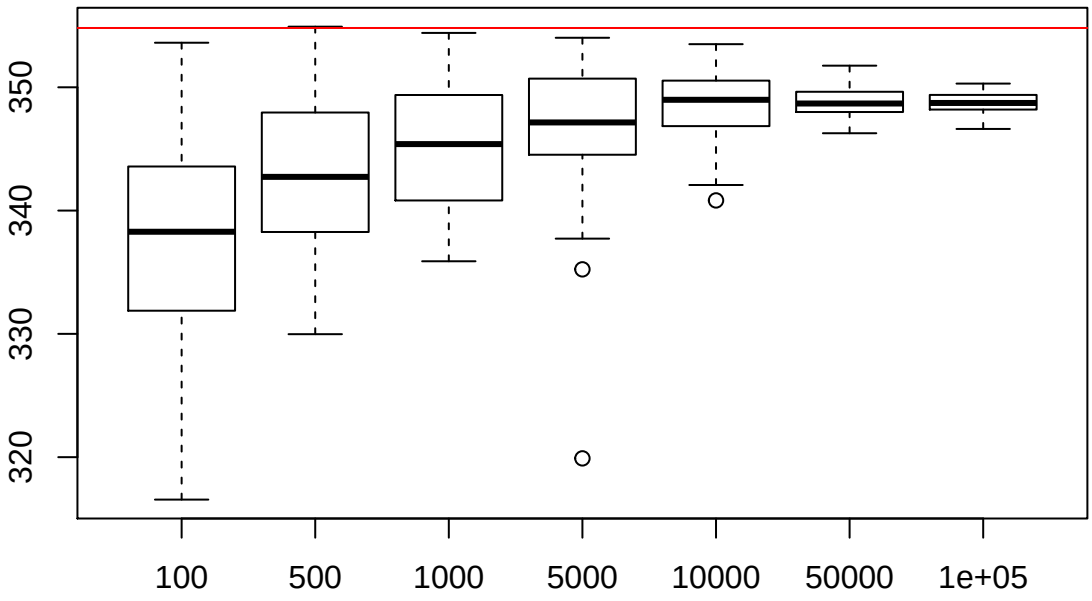
Log likelihood approximations

Warning: this takes a few minutes.

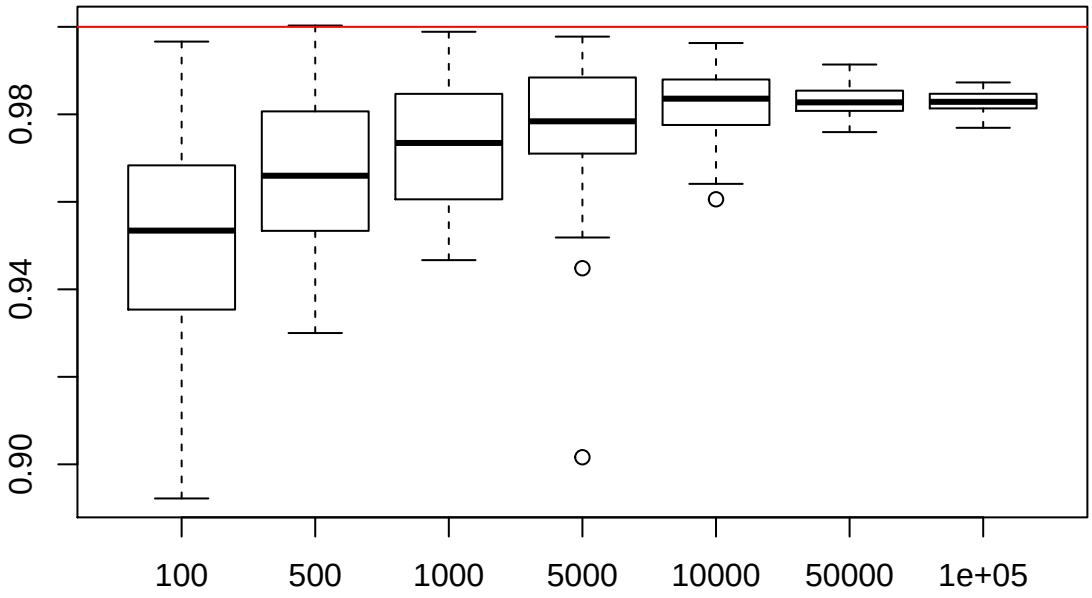
```
# storage list
results <- vector('list', length(nsamp))

# main loop
for (i in 1:length(nsamp)) {
  # cat('i =', i, '\n')
  r <- rep(NA, M)
  for (j in 1:M) {
    # cat('\tj =', j, '\n')
    # running into a rare problem where the estimated covariance matrix is not
    # positive definite. just replacing with NA here. only happens ~4 times
    # out of 700
    r[j] <- tryCatch(
      normal_ll(observations, dist_mat, nu, alpha, sigma, tau, num_samps = nsamp[i]),
      error = function(e) NA_real_
    )
  }
  results[[i]] <- r
}

# convert results to data frame for easier plotting
resultsdf <- data.frame(mc_size = rep(nsamp, each = M), ll = unlist(results))
# standardizing the log-likelihood values by dividing by the exact ll
resultsdf$ll_std <- resultsdf$ll / exact
```

Plots

```
# raw values
boxplot(ll ~ mc_size, data = resultsdf)
abline(h = exact, col = 'red')
```




```
# standardized values
boxplot(ll_std ~ mc_size, data = resultsdf)
abline(h = 1, col = 'red')
```



Accuracy does not improve with more than ~10,000 or so MC samples. There is a consistent bias of about -2% for some reason.