INF626L: Lab 1

Jun Rao

8/31/2019

Task 1 (5 points)

Power Function

To make this a bit easier, assume x is a real number and y is an integer. Use a for-loop. Do not use the base (^) function.

```
power_func = function(x,y){

  result = 1

  for(i in 1:y){
    result = result * x
  }

  return(result)
}

# Test my function:
x=2
y=10

power_func(x,y)

## [1] 1024
```

Task 2 (10 points)

Now, we'll start to familiarize ourselves with using Stan.

```
# Load the rstan package
library(rstan)

# Set some useful options
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

Now, I will alter the func_sim.stan file

```
x=2
y=10
sim_data = list(x = x,y = y)
sim_fit =
  stan(file="fun.stan",
  data=sim_data,
  iter=1,
  chains=1,
  algorithm="Fixed_param")

##
## SAMPLING FOR MODEL 'fun' NOW (CHAIN 1).
## Chain 1: Iteration: 1 / 1 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0 seconds (Warm-up)
## Chain 1:                0 seconds (Sampling)
## Chain 1:                0 seconds (Total)
## Chain 1:

# The function extract() will create a named list of output
# Note that "lp__" is irrelevant here, but will become
# very important later in the course.
sim_out = extract(sim_fit)
# View the structure of the sim_out object:
str(sim_out)

## List of 2
##  $ summation: num [1(1d)] 1024
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ iterations: NULL
##  $ lp__     : num [1(1d)] 0
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ iterations: NULL

# Call the value 'summation':
sim_out$summation

## [1] 1024

# Compare with our example function in Task 1:
power_func(x,y)

## [1] 1024

# Compare with R's base sum function
x=2
y=10
x^y

## [1] 1024
```