

Project 2

Code ▾

Hide

```
load("../data/datasets.RData")

str(tree_data)
```

```
'data.frame':  28 obs. of  3 variables:
 $ R_mean: num  0.601 0.557 0.506 0.506 0.506 ...
 $ G_mean: num  0.524 0.471 0.457 0.457 0.457 ...
 $ B_mean: num  0.438 0.391 0.379 0.379 0.379 ...
```

Hide

```
length(tree_labels)
```

```
[1] 28
```

Hide

```
install.packages("optimx")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.4/optimx_2025-4.9.tgz'
Content type 'application/x-gzip' length 2651187 bytes (2.5 MB)
=====
downloaded 2.5 MB
```

The downloaded binary packages are in
/var/folders/4h/1shgwhjj6895c05ncfv_9c2r0000gn/T//RtmpaQGCfC/downloaded_packages

Hide

```

library(optimx)

# Define logistic loss
compute_loss <- function(X, y, theta, loss_type) {
  scores <- X %*% theta
  probs <- 1 / (1 + exp(-scores))
  eps <- 1e-6

  if (loss_type == "logistic") {
    return(mean(log(1 + exp(-y * scores))))
  } else if (loss_type == "cross-entropy") {
    return(-mean(y * log(probs + eps) + (1 - y) * log(1 - probs + eps)))
  }
}

# Define the training function
train_optimx <- function(X, y, method = "BFGS", loss_type = "logistic") {
  X <- as.matrix(X)
  y <- as.numeric(y)
  theta_init <- rep(0, ncol(X))

  # Loss function for optimizer
  loss_fn <- function(theta) {
    compute_loss(X, y, theta, loss_type)
  }

  result <- optimx(par = theta_init, fn = loss_fn, method = method)
  return(result)
}

```

Hide

```

result_bfgs <- train_optimx(tree_data, tree_labels, method = "BFGS", loss_type = "log
istic")

```

Warning: no non-missing arguments to max; returning -Inf
Warning: no non-missing arguments to min; returning Inf

Hide

```

print(result_bfgs)

```

| p1 | p2 | p3 | value | fevals | gevals | niter | convcode | kkt1 |
|-------|-------|-------|-------|--------|--------|-------|----------|------|
| <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <lg> |

| | | | | | | | | | |
|------|---------|----------|----------|-----------|----|----|----|---|------|
| BFGS | 7.69712 | 6.709273 | 5.801079 | 0.2723392 | 16 | 15 | NA | 0 | TRUE |
|------|---------|----------|----------|-----------|----|----|----|---|------|

1 row | 1-10 of 11 columns

Hide

```

methods <- c("BFGS", "L-BFGS-B", "CG", "Nelder-Mead")

results_tree <- lapply(methods, function(m) {
  cat("Training Tree label with", m, "...\\n")
  train_optimx(tree_data, tree_labels, method = m, loss_type = "logistic")
})

```

Training Tree label with BFGS ...

Warning: no non-missing arguments to max; returning -Inf
Warning: no non-missing arguments to min; returning Inf

Training Tree label with L-BFGS-B ...

Warning: no non-missing arguments to max; returning -Inf
Warning: no non-missing arguments to min; returning Inf

Training Tree label with CG ...

Warning: no non-missing arguments to max; returning -Inf
Warning: no non-missing arguments to min; returning Inf

Training Tree label with Nelder-Mead ...

Warning: no non-missing arguments to max; returning -Inf
Warning: no non-missing arguments to min; returning Inf

Hide

```
names(results_tree) <- methods
```

Hide

```
sapply(results_tree, function(res) res$value)
```

| BFGS | L-BFGS-B | CG | Nelder-Mead |
|-----------|-----------|-----------|-------------|
| 0.2723392 | 0.2723078 | 0.2795314 | 0.2723078 |

Hide

```
# Compare final loss values from each optimizer
final_losses <- sapply(results_tree, function(res) res$value)
print(final_losses)
```

| BFGS | L-BFGS-B | CG | Nelder-Mead |
|-----------|-----------|-----------|-------------|
| 0.2723392 | 0.2723078 | 0.2795314 | 0.2723078 |

Hide

```
run_optim_with_tracking <- function(X, y, method = "BFGS", loss_type = "logistic") {
  X <- as.matrix(X)
  y <- as.numeric(y)
  theta <- rep(0, ncol(X))
  loss_values <- numeric(100)

  for (i in 1:100) {
    res <- optimx::optimx(par = theta, fn = function(th) compute_loss(X, y, th, loss_type),
                          method = method, control = list(maxit = 1))
    theta <- as.numeric(res[1, 1:ncol(X)]) # update theta
    loss_values[i] <- res$value
  }

  return(loss_values)
}

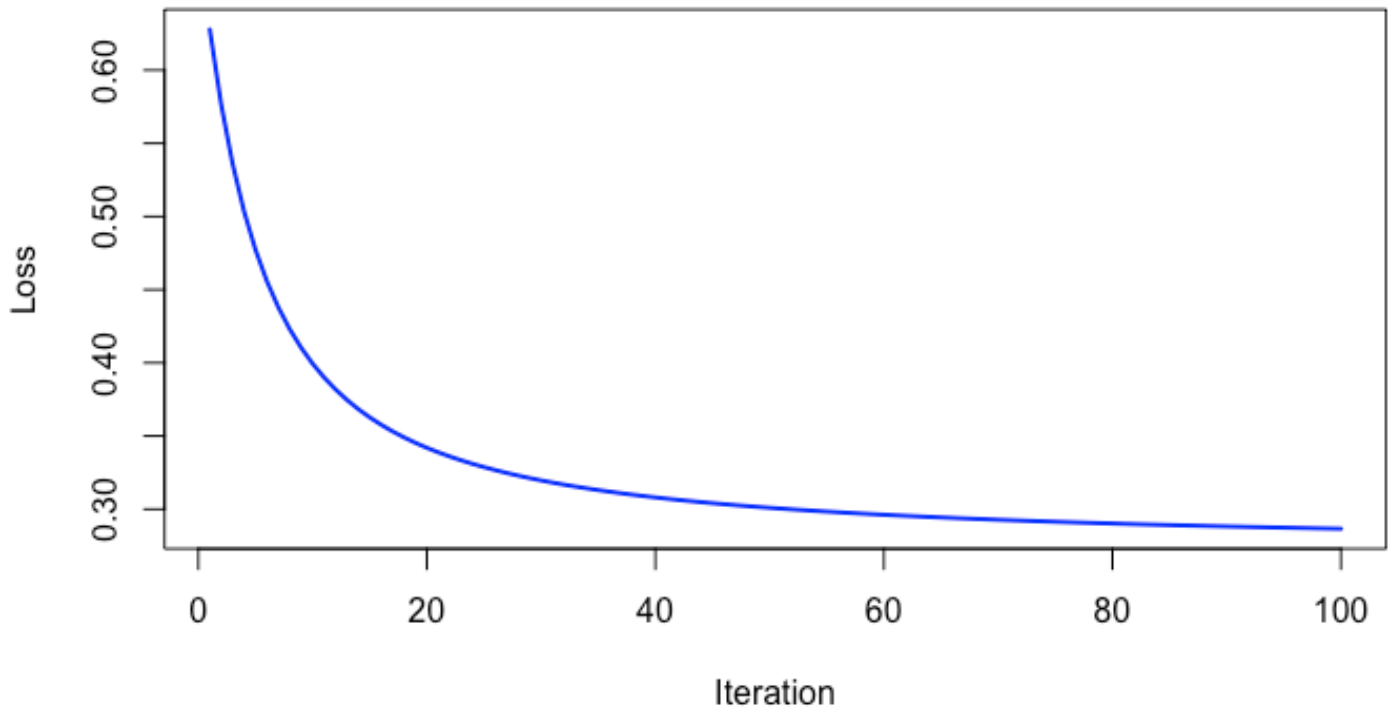
# Example for BFGS
loss_bfgs <- run_optim_with_tracking(tree_data, tree_labels, method = "BFGS")
```

Warning: no non-missing arguments to max; returning -Inf
Warning: no non-missing arguments to min; returning Inf

Hide

```
plot(loss_bfgs, type = "l", col = "blue", lwd = 2,
     main = "Loss Curve - BFGS", xlab = "Iteration", ylab = "Loss")
```

Loss Curve - BFGS

[Hide](#)

```
run_optim_with_tracking <- function(X, y, method = "BFGS", loss_type = "logistic", steps = 100) {  
  X <- as.matrix(X)  
  y <- as.numeric(y)  
  theta <- rep(0, ncol(X))  
  loss_values <- numeric(steps)  
  
  for (i in 1:steps) {  
    res <- optimx::optimx(  
      par = theta,  
      fn = function(th) compute_loss(X, y, th, loss_type),  
      method = method,  
      control = list(maxit = 1)  
    )  
    theta <- as.numeric(res[1, 1:ncol(X)])  
    loss_values[i] <- res$value  
  }  
  
  return(loss_values)  
}
```

Hide

```
loss_bfgs      <- run_optim_with_tracking(tree_data, tree_labels, method = "BFGS")
```

```
Warning: no non-missing arguments to max; returning -InfWarning: no non-missing argum
ents to min; returning Inf
```

Hide

```
loss_lbfgsb <- run_optim_with_tracking(tree_data, tree_labels, method = "L-BFGS-  
B")
```

```
Warning: no non-missing arguments to max; returning -InfWarning: no non-missing argum
ents to min; returning Inf
```

Hide

```
loss_cg      <- run_optim_with_tracking(tree_data, tree_labels, method = "CG")
```

[illegible]

[illegible]

[illegible]


```

arning: no non-missing arguments to max; returning -InfWarning: no non-missing argumen
ts to min; returning InfWarning: no non-missing arguments to max; returning -InfWarni
ng: no non-missing arguments to min; returning InfWarning: no non-missing arguments t
o max; returning -InfWarning: no non-missing arguments to min; returning InfWarning:
no non-missing arguments to max; returning -InfWarning: no non-missing arguments to m
in; returning InfWarning: no non-missing arguments to max; returning -InfWarning: no
non-missing arguments to min; returning InfWarning: no non-missing arguments to max;
returning -InfWarning: no non-missing arguments to min; returning InfWarning: no non-
missing arguments to max; returning -InfWarning: no non-missing arguments to min; ret
urning InfWarning: no non-missing arguments to max; returning -InfWarning: no non-mis
sing arguments to min; returning Inf

```

Hide

```

loss_neldermead <- run_optim_with_tracking(tree_data, tree_labels, method = "Nelder-
Mead")

```

```

Warning: no non-missing arguments to max; returning -InfWarning: no non-missing argum
ents to min; returning Inf

```

Hide

```

plot(loss_bfgs, type = "l", col = "blue", lwd = 2,
      xlab = "Iteration", ylab = "Loss", main = "Optimizer Loss Comparison", ylim = ra
nge(c(loss_bfgs, loss_lbfgsb, loss_cg, loss_neldermead)))
lines(loss_lbfgsb, col = "red", lwd = 2)

```

Hide

```

lines(loss_cg, col = "green", lwd = 2)
lines(loss_neldermead, col = "purple", lwd = 2)

```

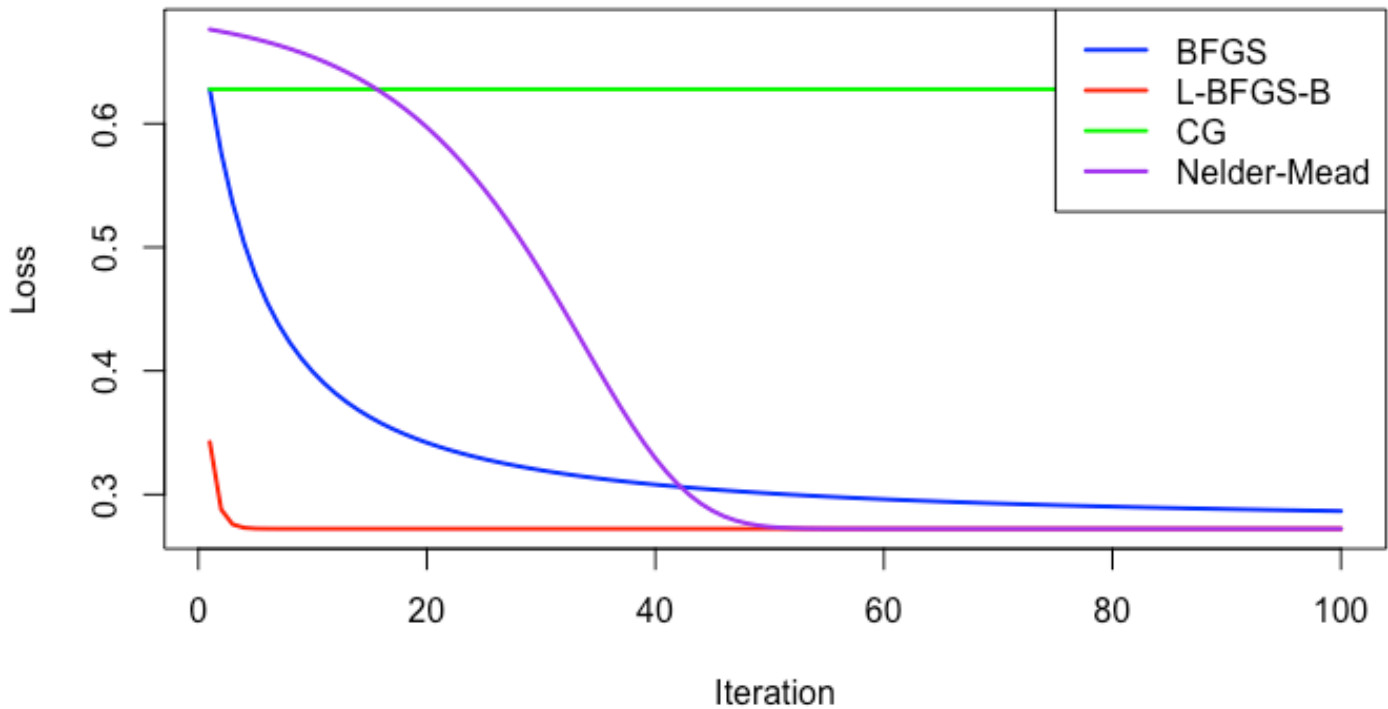
Hide

```

legend("topright", legend = c("BFGS", "L-BFGS-B", "CG", "Nelder-Mead"),
      col = c("blue", "red", "green", "purple"), lwd = 2)

```

Optimizer Loss Comparison

[Hide](#)

```
loss_bfgs_myth      <- run_optim_with_tracking(myth_data, myth_labels, method = "BFGS")
```

```
Warning: no non-missing arguments to max; returning -InfWarning: no non-missing arguments to min; returning Inf
```

[Hide](#)

```
loss_lbfgsb_myth    <- run_optim_with_tracking(myth_data, myth_labels, method = "L-BFGS-B")
```

```
Warning: no non-missing arguments to max; returning -InfWarning: no non-missing arguments to min; returning Inf
```

[Hide](#)

```
loss_cg_myth        <- run_optim_with_tracking(myth_data, myth_labels, method = "CG")
```

[illegible]

[illegible]

```
no non-missing arguments to max; returning -InfWarning: no non-missing arguments to m
in; returning InfWarning: no non-missing arguments to max; returning -InfWarning: no
non-missing arguments to min; returning InfWarning: no non-missing arguments to max;
returning -InfWarning: no non-missing arguments to min; returning InfWarning: no non-
missing arguments to max; returning -InfWarning: no non-missing arguments to min; ret
urning InfWarning: no non-missing arguments to max; returning -InfWarning: no non-mis
sing arguments to min; returning InfWarning: no non-missing arguments to max; returni
ng -InfWarning: no non-missing arguments to min; returning InfWarning: no non-missing
arguments to max; returning -InfWarning: no non-missing arguments to min; returning I
nfWarning: no non-missing arguments to max; returning -InfWarning: no non-missing arg
uments to min; returning InfWarning: no non-missing arguments to max; returning -InfW
arning: no non-missing arguments to min; returning InfWarning: no non-missing argumen
ts to max; returning -InfWarning: no non-missing arguments to min; returning InfWarni
ng: no non-missing arguments to max; returning -InfWarning: no non-missing arguments
to min; returning InfWarning: no non-missing arguments to max; returning -InfWarning:
no non-missing arguments to min; returning InfWarning: no non-missing arguments to ma
x; returning -InfWarning: no non-missing arguments to min; returning InfWarning: no n
on-missing arguments to max; returning -InfWarning: no non-missing arguments to min;
returning InfWarning: no non-missing arguments to max; returning -InfWarning: no non-
missing arguments to min; returning InfWarning: no non-missing arguments to max; retu
rning -InfWarning: no non-missing arguments to min; returning InfWarning: no non-miss
ing arguments to max; returning -InfWarning: no non-missing arguments to min; returni
ng InfWarning: no non-missing arguments to max; returning -InfWarning: no non-missing
arguments to min; returning InfWarning: no non-missing arguments to max; returning -I
nfWarning: no non-missing arguments to min; returning InfWarning: no non-missing argu
ments to max; returning -InfWarning: no non-missing arguments to min; returning InfWa
rning: no non-missing arguments to max; returning -InfWarning: no non-missing argumen
ts to min; returning InfWarning: no non-missing arguments to max; returning -InfWarni
ng: no non-missing arguments to min; returning InfWarning: no non-missing arguments t
o max; returning -InfWarning: no non-missing arguments to min; returning InfWarning:
no non-missing arguments to max; returning -InfWarning: no non-missing arguments to m
in; returning InfWarning: no non-missing arguments to max; returning -InfWarning: no
non-missing arguments to min; returning InfWarning: no non-missing arguments to max;
returning -InfWarning: no non-missing arguments to min; returning InfWarning: no non-
missing arguments to max; returning -InfWarning: no non-missing arguments to min; ret
urning InfWarning: no non-missing arguments to max; returning -InfWarning: no non-mis
sing arguments to min; returning Inf
```

[Hide](#)

```
loss_neldermead_myth <- run_optim_with_tracking(myth_data, myth_labels, method = "Nel
der-Mead")
```

```
Warning: no non-missing arguments to max; returning -InfWarning: no non-missing argum
ents to min; returning Inf
```

[Show](#)

Hide

```
plot(loss_bfgs_myth, type = "l", col = "blue", lwd = 2,  
     xlab = "Iteration", ylab = "Loss", main = "Mythology - Optimizer Loss Comparison",  
     ylim = range(c(loss_bfgs_myth, loss_lbfgsb_myth, loss_cg_myth, loss_neldermead_myth)))  
lines(loss_lbfgsb_myth, col = "red", lwd = 2)
```

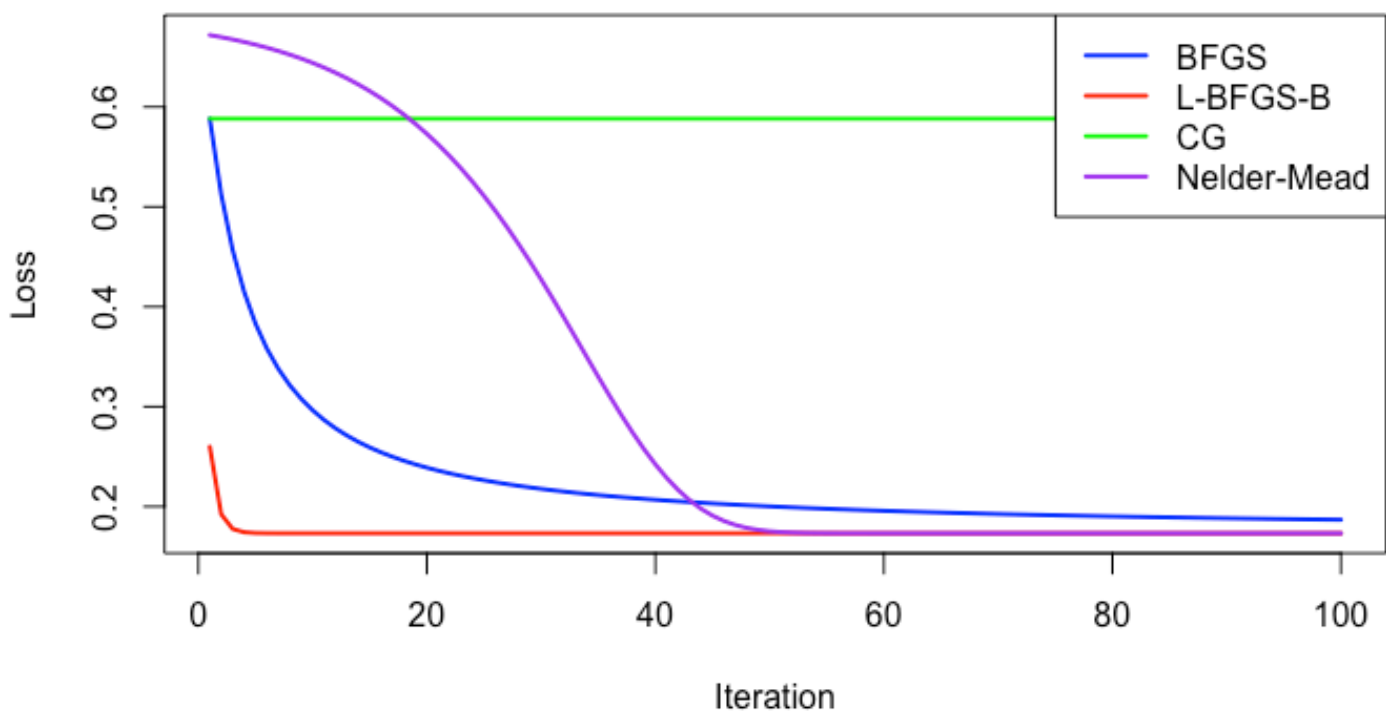
Hide

```
lines(loss_cg_myth, col = "green", lwd = 2)  
lines(loss_neldermead_myth, col = "purple", lwd = 2)
```

Hide

```
legend("topright", legend = c("BFGS", "L-BFGS-B", "CG", "Nelder-Mead"),  
     col = c("blue", "red", "green", "purple"), lwd = 2)
```

Mythology - Optimizer Loss Comparison



Hide

```
Warning: no non-missing arguments to max; returning -InfWarning: no non-missing argum
ents to min; returning Inf
```

```
Warning: no non-missing arguments to max; returning -InfWarning: no non-missing argum
ents to min; returning Inf
```

[illegible]

[illegible]

[illegible]

Hide

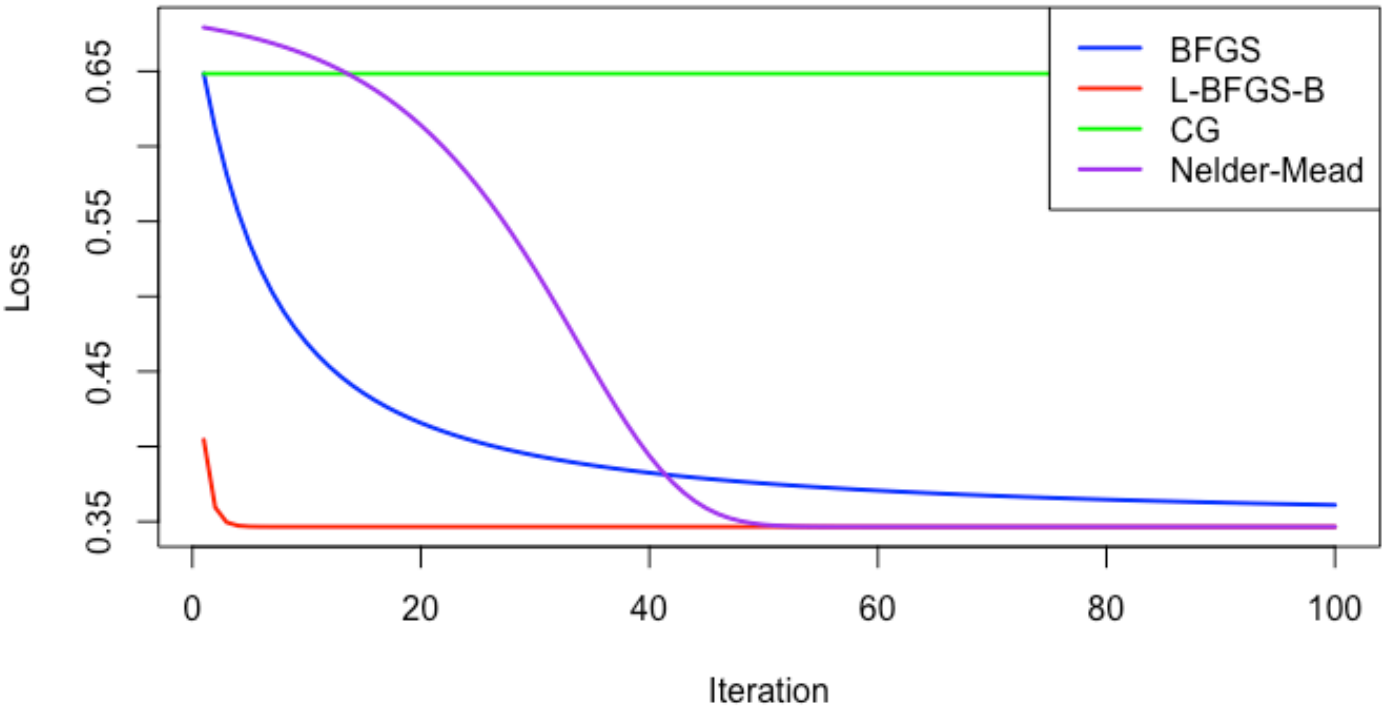
Hide

Hide

Hide

Page 18 of 19

Animal - Optimizer Loss Comparison



Hide

```
# Combine results
summary_df <- data.frame(
  Method = names(results_tree),
  Final_Loss = sapply(results_tree, function(res) res$value)
)
print(summary_df)
```

| | Method<chr> | Final_Loss<dbl> |
|-------------|-------------|-----------------|
| BFGS | BFGS | 0.2723392 |
| L-BFGS-B | L-BFGS-B | 0.2723078 |
| CG | CG | 0.2795314 |
| Nelder-Mead | Nelder-Mead | 0.2723078 |

4 rows