DS assessment question 2

Jianing Yao

2023-08-26

```r
# Check if two inputs are close enough
are_close <- function(v, w, abs_tol = 1e-6, rel_tol = 1e-6) {
  abs_diff <- abs(v - w)
  are_all_within_atol <- all(abs_diff < abs_tol)
  are_all_within_rtol <- all(abs_diff < rel_tol * pmax(abs(v), abs(w)))
  return(are_all_within_atol && are_all_within_rtol)
}
```

```r
# Calculate gradient of L(b) = ||y-bx||^2
grad <- function(x, y, b) {
  gradient <- - 2 * sum(x * (y - b * x))
  return(gradient)
}
```

```r
# Function for solving a simple linear regression using gradient descent
gd_solver <- function(x, y, learning_rate = 1e-3, iterations = 1e+3, tol = 1e-6){
  b <- 0
  converged <- FALSE
  iter <- 1
  while ((iter < iterations) && (!converged)) {
    b_old <- b
    gradient <- grad(x, y, b)
    b <- b - learning_rate * gradient
    converged <- are_close(b, b_old, abs_tol = tol, rel_tol = tol)
    iter <- iter + 1
  }
  if (converged) {
      cat("b is found after ", iter - 1, " iterations with convergence. Estimated b is ", b, "\n")
  } else {
      warning("gd solver failed to converge after ", iter - 1, " iterations.
            The estimates may be meaningless. Estimated b is ", b, "\n")
  }
  return(b)
}
```

```r
# Test gd_solver on some randomly generated normal vectors
set.seed(100)
n <- 10
x <- rnorm(n)
b_true <- 5 # true value of b
y <- b_true * x
```

```r
num_iterations <- 10000
learning_rate <- 1e-3
```

```r
b_est <- gd_solver(x, y, learning_rate = learning_rate, iterations = num_iterations, tol = 1e-8)
```

```
## b is found after  2613  iterations with convergence. Estimated b is  4.999998
```

```r
num_iterations <- 10000
learning_rate <- 1e-2
b_est <- gd_solver(x, y, learning_rate = learning_rate, iterations = num_iterations, tol = 1e-8)
```

```
## b is found after  295  iterations with convergence. Estimated b is  5
```

```r
num_iterations <- 10000
learning_rate <- 0.5
b_est <- gd_solver(x, y, learning_rate = learning_rate, iterations = num_iterations, tol = 1e-8)
```

```
## Error in while ((iter < iterations) && (!converged)) {: missing value where TRUE/FALSE needed
```

How does the performance of the algorithm's depend on e? When does the algorithm fail and why?

When the learning rate is small, the algorithm takes longer time to converge (i.e, 2613 iterations for lr $= 0.001$ vs 295 iterations for lr $= 0.01$). However, when the learning rate is too high, the algorithm may overshoot the minimum of the loss function and it may oscillate around the true value or go to infinity. In this case, the algorithm never converges.