

## Multivariate Optimization 1

### Exercise 1: Gradient Descent

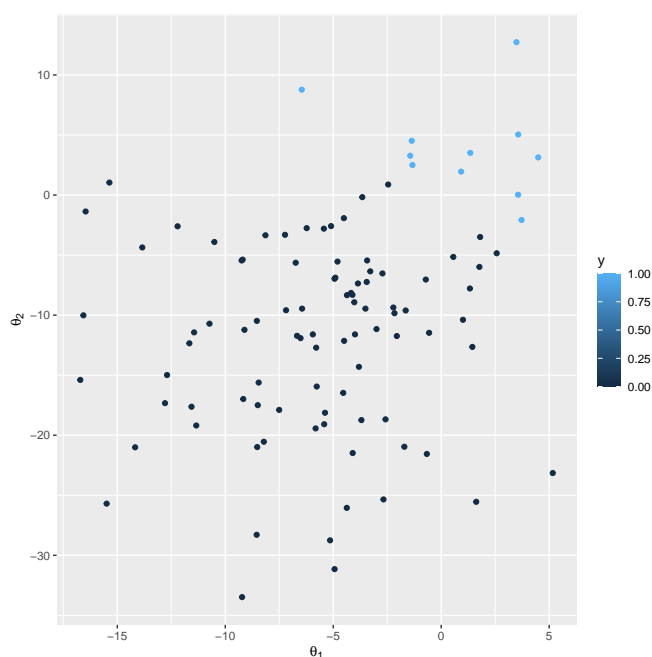
You are given the following data situation:

```
library(ggplot2)

set.seed(314)
n <- 100
X = cbind(rnorm(n, -5, 5),
          rnorm(n, -10, 10))
X_design = cbind(1, X)

z <- 2*X[,1] + 3*X[,2]
pr <- 1/(1+exp(-z))
y <- as.integer(pr > 0.5)
df <- data.frame(X = X, y = y)

ggplot(df) +
  geom_point(aes(x = X[,1], y = X[,2], color=y)) +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))
```



In the following we want to estimate a logistic regression without intercept via gradient descent<sup>1</sup>.

- (a) First consider the derivative of  $g : \mathbb{R} \rightarrow \mathbb{R}, z \mapsto \log(1 + \exp(z)) - z$ , i.e.,

$$g'(z) = \underbrace{\frac{\exp(z)}{1 + \exp(z)}}_{<1} - 1 < 0 \Rightarrow g \text{ is monotonically decreasing} \Rightarrow g(z) > g(\alpha z) \quad \forall z > 0 \text{ and } \alpha > 0.$$

<sup>1</sup>We chose this algorithm for educational purposes; in practice, we typically use second order algorithms.

Second consider the derivative of  $h : \mathbb{R} \rightarrow \mathbb{R}, z \mapsto \log(1 + \exp(-z))$ , i.e.,

$$h'(z) = - \underbrace{\frac{\exp(-z)}{1 + \exp(-z)}}_{>0} < 0 \Rightarrow h \text{ is monotonically decreasing} \Rightarrow h(z) > h(\alpha z) \quad \forall z > 0 \text{ and } \alpha > 0.$$

With this we get for  $\alpha > 0$

$$\begin{aligned} \mathcal{R}_{\text{emp}}(\tilde{\theta}) &= \sum_{i=1}^n \log(1 + \exp(\tilde{\theta}^\top \mathbf{x}^{(i)})) - y^{(i)} \tilde{\theta}^\top \mathbf{x}^{(i)} = \\ &= \sum_{i=1}^n \mathbb{1}_{y^{(i)}=1} (\log(1 + \exp(|\tilde{\theta}^\top \mathbf{x}^{(i)}|)) - |\tilde{\theta}^\top \mathbf{x}^{(i)}|) + \mathbb{1}_{y^{(i)}=0} (\log(1 + \exp(-|\tilde{\theta}^\top \mathbf{x}^{(i)}|)) - (-|\tilde{\theta}^\top \mathbf{x}^{(i)}|)) > \\ &= \sum_{i=1}^n \mathbb{1}_{y^{(i)}=1} (\log(1 + \exp(\alpha |\tilde{\theta}^\top \mathbf{x}^{(i)}|)) - \alpha |\tilde{\theta}^\top \mathbf{x}^{(i)}|) + \mathbb{1}_{y^{(i)}=0} (\log(1 + \exp(-\alpha |\tilde{\theta}^\top \mathbf{x}^{(i)}|)) - (-\alpha |\tilde{\theta}^\top \mathbf{x}^{(i)}|)) = \\ &= \sum_{i=1}^n \log(1 + \exp(\alpha \tilde{\theta}^\top \mathbf{x}^{(i)})) - y^{(i)} \alpha \tilde{\theta}^\top \mathbf{x}^{(i)} = \\ &= \mathcal{R}_{\text{emp}}(\alpha \tilde{\theta}) \text{ since } \tilde{\theta} \text{ perfectly separates the data.} \end{aligned}$$

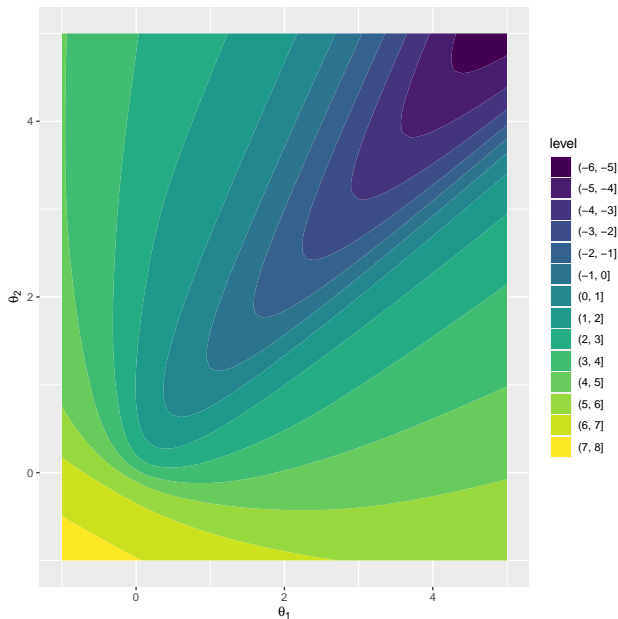
(b) lambda = 0

```
f <- function(theta, lambda) lambda * theta %*% theta +
  sum(-y * X %*% theta + log(1 + exp(X %*% theta)))

x = seq(-1, 5, by=0.1)
xx = expand.grid(X1 = x, X2 = x)

fxx = log(apply(xx, 1, function(t) f(t, lambda)))
df = data.frame(xx = xx, fxx = fxx)

ggplot() +
  geom_contour_filled(data = df, aes(x = xx.X1, y = xx.X2, z = fxx)) +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))
```



$$(c) \quad \frac{\partial}{\partial \theta} \mathcal{R}_{\text{emp}} = \sum_{i=1}^n \frac{\exp(\theta^\top \mathbf{x}^{(i)})}{1 + \exp(\theta^\top \mathbf{x}^{(i)})} \mathbf{x}^{(i)\top} - y^{(i)} \mathbf{x}^{(i)\top}$$

```
df_t <- function(theta, lambda) lambda * t(theta) - (t(y) %*% X) +
  t(1/(1 + exp(-X %*% theta))) %*% X

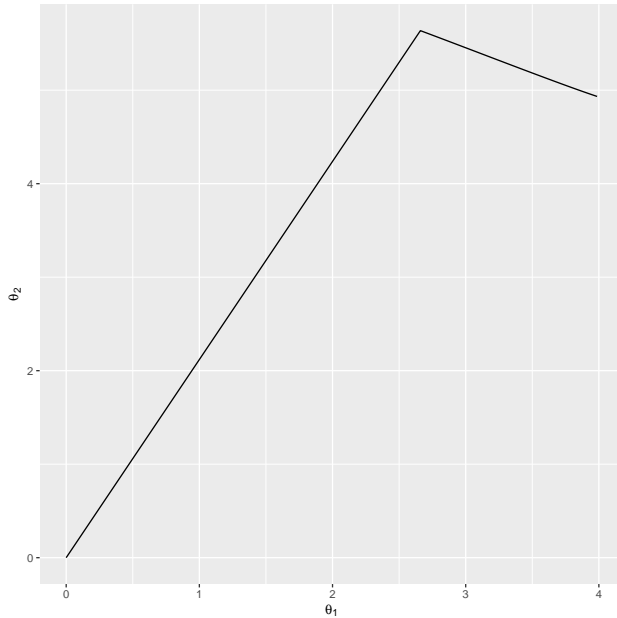
gd_step <- function(theta, alpha, lambda) return(theta - alpha * df_t(theta, lambda)[1,])
## Alpha = 0.5
theta = c(0,0)
alpha = 0.01
```

```

thetas = theta
for(i in 1:500){
  theta = gd_step(theta, alpha, lambda)
  thetas = rbind(thetas, theta)
}

ggplot(as.data.frame(thetas), aes(x=V1, y=V2)) +
  geom_line() +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))

```



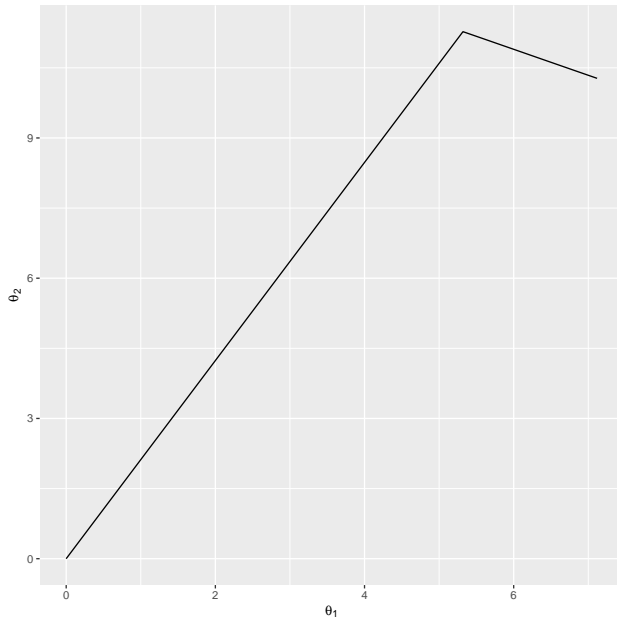
```

## Alpha = 2
theta = c(0,0)
alpha = 0.02

thetas = theta
for(i in 1:500){
  theta = gd_step(theta, alpha, lambda)
  thetas = rbind(thetas, theta)
}

ggplot(as.data.frame(thetas), aes(x=V1, y=V2)) +
  geom_line() +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))

```

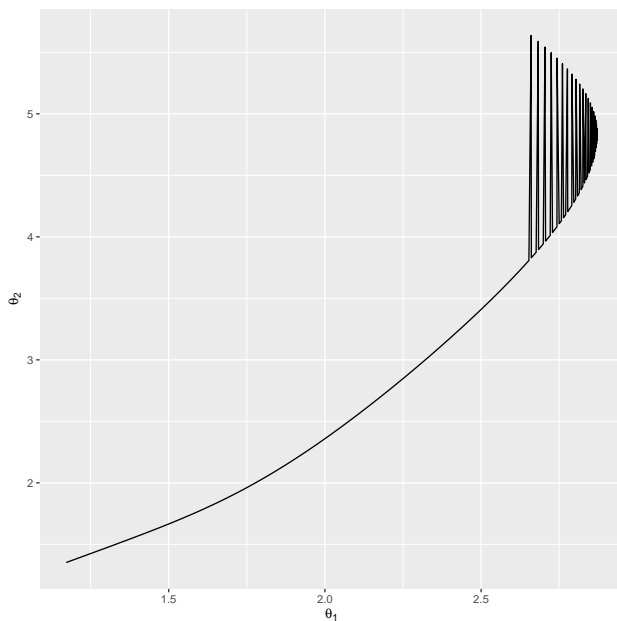


Gradient descent will in theory not converge since  $\mathcal{R}_{\text{emp}}$  has no minimum (a))

```
(e) ## Lambda = 0.5, alpha = 0.5
theta = c(0,0)
alpha = 0.01

thetas = NULL
for(i in 1:500){
  theta = gd_step(theta, alpha, 0.5)
  thetas = rbind(thetas, theta)
}

ggplot(as.data.frame(thetas), aes(x=V1, y=V2)) +
  geom_line() +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))
```



```

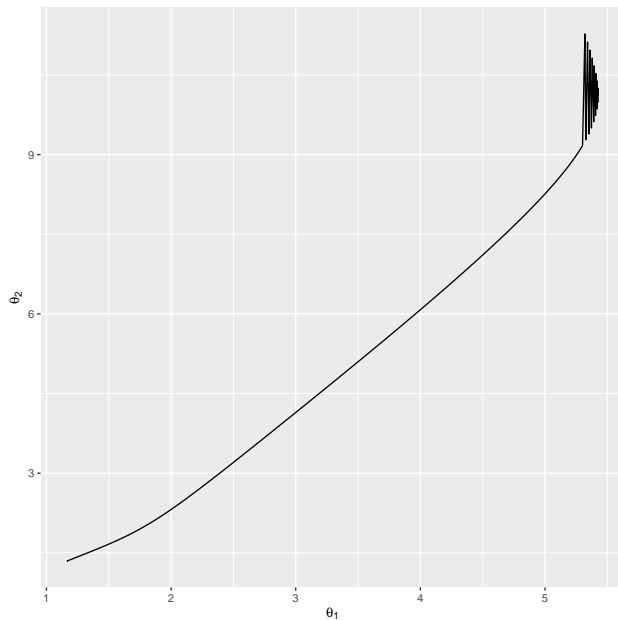
theta = c(0,0)
alpha = 0.02

thetas = theta
for(i in 1:500){
  theta = gd_step(theta, alpha, 0.5)
  thetas = rbind(thetas, theta)
}
## Lambda = 0.5, alpha = 0.02
theta = c(0,0)
alpha = 0.02

thetas = NULL
for(i in 1:500){
  theta = gd_step(theta, alpha, 0.5)
  thetas = rbind(thetas, theta)
}

ggplot(as.data.frame(thetas), aes(x=V1, y=V2)) +
  geom_line() +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))

```



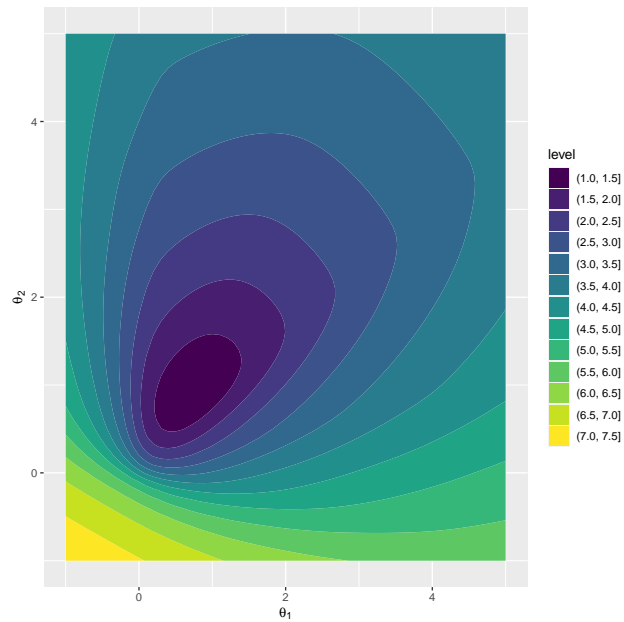
(f) `lambda = 1`

```

fxx_reg = log(apply(xx, 1, function(t) f(t, lambda)))
df_reg = data.frame(xx = xx, fxx = fxx_reg)

ggplot() +
  geom_contour_filled(data = df_reg, aes(x = xx.X1, y = xx.X2, z = fxx)) +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))

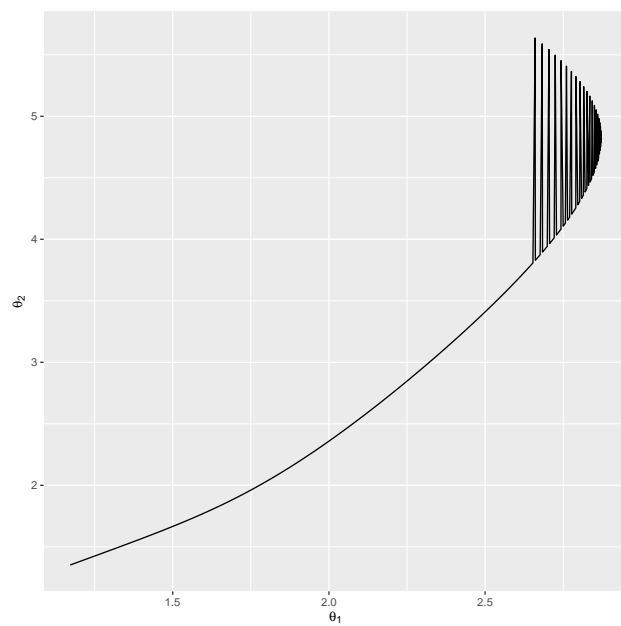
```



```
(g) ## Lambda = 0.5, alpha = 0.5
theta = c(0,0)
alpha = 0.01

thetas = NULL
for(i in 1:500){
  theta = gd_step(theta, alpha, 0.5)
  thetas = rbind(thetas, theta)
}

ggplot(as.data.frame(thetas), aes(x=V1, y=V2)) +
  geom_line() +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))
```



```

theta = c(0,0)
alpha = 0.02

thetas = theta
for(i in 1:500){
  theta = gd_step(theta, alpha, 0.5)
  thetas = rbind(thetas, theta)
}
## Lambda = 0.5, alpha = 0.02
theta = c(0,0)
alpha = 0.02

thetas = NULL
for(i in 1:500){
  theta = gd_step(theta, alpha, 0.5)
  thetas = rbind(thetas, theta)
}

ggplot(as.data.frame(thetas), aes(x=V1, y=V2)) +
  geom_line() +
  xlab(expression(theta[1])) +
  ylab(expression(theta[2]))

```

