

Nonlinear Programming 1

Solution 1: Lagrange multipliers

We know that at a minimum (x^*, y^*) the contour lines of the objective function f and the equality constraint g touch and thus it needs to hold that

$$\nabla f(x^*, y^*) = \lambda \nabla g(x^*, y^*)$$

for some Lagrange multiplier $\lambda \in \mathbb{R}$.

Hence, we get the two equalities

$$1 = \lambda 2x^*$$

$$2 = \lambda 8y^*$$

yielding $x^* = 2y^*$.

The third equality is the equality constraint. It gives

$$\begin{aligned} x^{*2} + 4y^{*2} &= 4 \\ \Leftrightarrow 4y^{*2} + 4y^{*2} &= 4 \\ \Leftrightarrow 8y^{*2} &= 4 \\ \Leftrightarrow y^{*2} &= \frac{1}{2}. \end{aligned}$$

Therefore, we have to candidate solutions

$$(x_1^*, y_1^*) = (\sqrt{2}, 1/\sqrt{2}) \quad \text{or} \quad (x_2^*, y_2^*) = (-\sqrt{2}, -1/\sqrt{2}).$$

Since $f(x_1^*, y_1^*) > f(x_2^*, y_2^*)$, we get the solution

$$(x^*, y^*) = (-\sqrt{2}, -1/\sqrt{2}).$$

Solution 2: Nonlinear SVM

(a)

$$\mathcal{L} = 0.5 \|\theta\|^2 + C \sum_{i=1}^n \zeta^{(i)} - \sum_{i=1}^n \alpha^{(i)} (\mathbf{y}^{(i)} (\phi(\mathbf{x}^{(i)})^\top \theta + \theta_0) - 1 + \zeta^{(i)}) - \sum_{i=1}^n \mu^{(i)} \zeta^{(i)}$$

(b)

```
library(ggplot2)
library(mlr3)
library(RColorBrewer)
library(CVXR)

##
## Attaching package: 'CVXR'
## The following object is masked from 'package:stats':
##
##     power

# generate 200 nonlinear separable binary observations
set.seed(123)
```

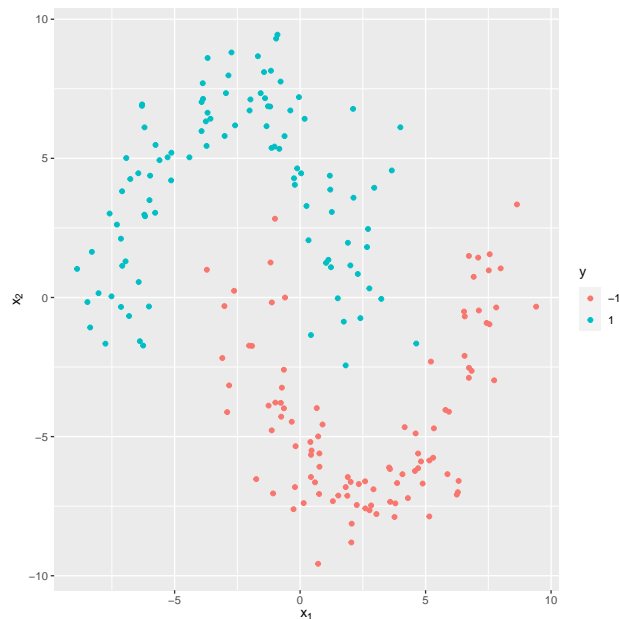
```

n = 200
moon_data = tgen("moons")$generate(n)$data()

moon_data$y = ifelse(moon_data$y == "A", 1, -1)
moon_data$y_dec = as.factor(moon_data$y)

ggplot(moon_data, aes(x=x1, y=x2)) +
  geom_point(aes(color=y_dec)) +
  xlab(expression(x[1])) +
  ylab(expression(x[2])) +
  labs(color=expression(y))

```



```

X = as.matrix(moon_data[,c("x1", "x2")])
# define cubic polynomial transformation (without intercept)
cubic_trafo <- function(x) return(cbind(x[1], x[2], x[1]^2, x[2]^2, x[1]*x[2],
                                         x[1]^2*x[2], x[1]*x[2]^2, x[1]^3,
                                         x[2]^3))

Z = t(apply(t(X), 2, cubic_trafo))
C = 1.0

# define variables for cvxr
theta0 = Variable()
theta = Variable(9)
slack = Variable(n)

objective = (1/2) * sum_squares(theta) + C * sum(slack)
constraints = list(moon_data$y * (Z %*% theta + theta0) >= 1 - slack, slack >= 0)
problem = Problem(Minimize(objective), constraints)
solution = solve(problem)

theta0_sol = solution$getValue(theta0)
theta_sol = solution$getValue(theta)

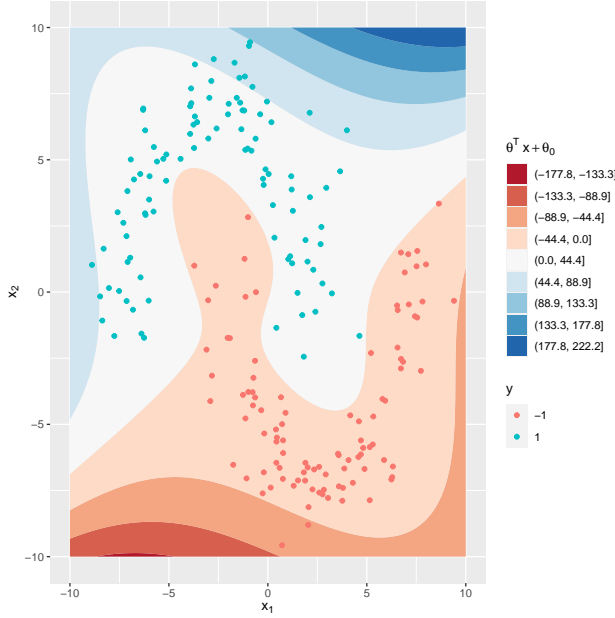
# create grid for plot
x = seq(-10, 10, by=0.05)
xx = expand.grid(X1 = x, X2 = x)
yxx = as.matrix(cubic_trafo(xx)) %*% theta_sol + theta0_sol

```

```

df = data.frame(xx = xx, yxx = yxx)
ggplot() +
  geom_contour_filled(data = df, aes(x = xx.X1, y = xx.X2, z = yxx), bins=10) +
  scale_fill_brewer(palette = "RdBu") +
  xlab(expression(x[1])) +
  ylab(expression(x[2])) +
  geom_point(data = moon_data, aes(x=x1, y=x2, color=y_dec)) +
  labs(color = expression(y), fill = expression(theta^T x + theta[0]))

```



(c) Stationarity

$$\nabla_{\theta} \mathcal{L} = \theta - \sum_{i=1}^n \alpha^{(i)} \mathbf{y}^{(i)} \phi(\mathbf{x}^{(i)}) = 0$$

$$\nabla_{\theta_0} \mathcal{L} = - \sum_{i=1}^n \alpha^{(i)} \mathbf{y}^{(i)} = 0$$

$$\nabla_{\zeta} \mathcal{L} = C \cdot \mathbf{1}_n - \alpha - \mu = 0$$

Primal feasibility

$$\begin{aligned}
-(\mathbf{y}^{(i)} (\phi(\mathbf{x}^{(i)})^\top \boldsymbol{\theta} + \theta_0) - 1 + \zeta^{(i)}) &\leq 0 \\
-\zeta^{(i)} &\leq 0
\end{aligned}$$

Dual feasibility

$$\mu \geq 0$$

$$\alpha \geq 0$$

Complementary slackness

$$\begin{aligned}
-\mu^{(i)} \zeta^{(i)} &= 0 \quad i = 1, \dots, n \\
-\alpha^{(i)} (\mathbf{y}^{(i)} (\phi(\mathbf{x}^{(i)})^\top \boldsymbol{\theta} + \theta_0) - 1 + \zeta^{(i)}) &= 0 \quad i = 1, \dots, n
\end{aligned}$$

(d) From the KKT conditions it follows that

$$\boldsymbol{\theta} = \sum_{i=1}^n \alpha^{(i)} \mathbf{y}^{(i)} \phi(\mathbf{x}^{(i)}),$$

$$\sum_{i=1}^n \alpha^{(i)} \mathbf{y}^{(i)} = 0,$$

$$C - \underbrace{\mu^{(i)}}_{\geq 0} = \alpha^{(i)} \Rightarrow C \geq \alpha^{(i)} \quad i = 1, \dots, n.$$

Plugging these into the Lagrangian gives

$$\begin{aligned} 0.5 \left\| \sum_{i=1}^n \alpha^{(i)} \mathbf{y}^{(i)} \phi(\mathbf{x}^{(i)}) \right\|^2 + \sum_{i=1}^n \mu^{(i)} \zeta^{(i)} + \sum_{i=1}^n \alpha^{(i)} \zeta^{(i)} - \left\| \sum_{i=1}^n \alpha^{(i)} \mathbf{y}^{(i)} \phi(\mathbf{x}^{(i)}) \right\|^2 + \sum_{i=1}^n \alpha^{(i)} - \sum_{i=1}^n \alpha^{(i)} \zeta^{(i)} - \sum_{i=1}^n \mu^{(i)} \zeta^{(i)} \\ = -0.5 \left\| \sum_{i=1}^n \alpha^{(i)} \mathbf{y}^{(i)} \phi(\mathbf{x}^{(i)}) \right\|^2 + \sum_{i=1}^n \alpha^{(i)} = -0.5 \sum_{i=1}^n \sum_{j=1}^n \alpha^{(i)} \alpha^{(j)} \mathbf{y}^{(i)} \mathbf{y}^{(j)} \langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \rangle + \sum_{i=1}^n \alpha^{(i)} \end{aligned}$$

Hence, the dual form of the nonlinear SVM is

$$\max_{\alpha} -0.5 \sum_{i=1}^n \sum_{j=1}^n \alpha^{(i)} \alpha^{(j)} \mathbf{y}^{(i)} \mathbf{y}^{(j)} \langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \rangle + \sum_{i=1}^n \alpha^{(i)}$$

s.t.

$$\sum_{i=1}^n \alpha^{(i)} \mathbf{y}^{(i)} = 0,$$

$$0 \leq \alpha \leq C.$$

Here, we can use the kernel trick to evaluate $\langle \phi(\mathbf{x}^{(i)}), \phi(\mathbf{x}^{(j)}) \rangle$ without explicitly computing the projections of each observation. (We only need to compute $\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle$)

```
(e) gram = (X %>% t(X) + 1)^3 - 1

alpha = Variable(n)
P = diag(moon_data$y) %>% gram %>% diag(moon_data$y)
P = P + diag(n)*0.0000001

objective = sum(alpha) - (1/2) * quad_form(alpha, P)
constraints = list(t(alpha) %>% moon_data$y == 0,
                  alpha >= 0,
                  C >= alpha)
problem = Problem(Maximize(objective), constraints)
solution = solve(problem)

solution$status

## [1] "optimal"

D = diag(c(sqrt(3), sqrt(3), sqrt(3), sqrt(3), sqrt(6), sqrt(3), sqrt(3), 1, 1))
theta_sol = c(t(solution$getValue(alpha) * moon_data$y) %>% Z %>% D)
theta0_sol = -0.5 * (max((z_m = Z %>% D %>% theta_sol)[moon_data$y == -1]) +
                  min((z_m = Z %>% D %>% theta_sol)[moon_data$y == 1]))

x = seq(-10, 10, by=0.05)
xx = expand.grid(X1 = x, X2 = x)
yxx = as.matrix(cubic_trafo(xx)) %>% D %>% theta_sol + theta0_sol

df = data.frame(xx = xx, yxx = yxx)
ggplot() +
  geom_contour_filled(data = df, aes(x = xx.X1, y = xx.X2, z = yxx), bins=10) +
  scale_fill_brewer(palette = "RdBu") +
  xlab(expression(x[1])) +
```

```
ylab(expression(x[2])) +
geom_point(data = moon_data, aes(x=x1, y=x2, color=y_dec)) +
labs(color = expression(y), fill = expression(theta^T x + theta[0]))
```

