Solution 1: Multiclass Hinge Loss

(a) We want to show that

$$L_{0-1}(y, h(\mathbf{x})) = \mathbb{1}_{\{y \neq h(\mathbf{x})\}} \leq L(y, f(\mathbf{x})) = \max_k \left(f_k(\mathbf{x}) - f_y(\mathbf{x}) + \mathbb{1}_{\{y \neq k\}} \right),$$

where

$$\mathcal{H} = \{ f = (f_1, \dots, f_g)^\top : \mathcal{X} \to \mathbb{R}^g \mid f_i : \mathcal{X} \to \mathbb{R}, \ \forall i \in \mathcal{Y} \}.$$

and

$$h(\mathbf{x}) = \arg\max_{k \in \{1,\dots,g\}} f_k(\mathbf{x}). \tag{1}$$

We distinguish two cases for any arbitrary data point pair $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$.

Case 1: $h(\mathbf{x}) \neq y$

Thus, $L_{0-1}(y, h(\mathbf{x})) = \mathbb{1}_{\{y \neq h(\mathbf{x})\}} = 1$ and in light of (1) this means that $y \neq \arg\max_{k \in \{1,\dots,g\}} f_k(\mathbf{x})$, so that there exists some $\tilde{k} \neq y$ with $f_{\tilde{k}}(\mathbf{x}) \geq f_y(\mathbf{x})$.

$$\begin{aligned}
L(y, f(\mathbf{x})) &= \max_k \left(f_k(\mathbf{x}) - f_y(\mathbf{x}) + \mathbb{1}_{\{y \neq k\}} \right) && \text{(Definition)} \\
&\geq \left(\underbrace{f_{\tilde{k}}(\mathbf{x}) - f_y(\mathbf{x})}_{\geq 0} + \mathbb{1}_{\{y \neq \tilde{k}\}} \right) && \text{(max always greater than one single component)} \\
&\geq \mathbb{1}_{\{y \neq \tilde{k}\}} && (f_{\tilde{k}}(\mathbf{x}) \geq f_y(\mathbf{x})) \\
&= 1. && \text{(Indicator function is true)}
\end{aligned}$$

Case 2: $h(\mathbf{x}) = y$

Thus, $L_{0-1}(y, h(\mathbf{x})) = \mathbb{1}_{\{y \neq h(\mathbf{x})\}} = 0$. But it always holds that

$$\begin{aligned}
L(y, f(\mathbf{x})) &= \max_k \left(f_k(\mathbf{x}) - f_y(\mathbf{x}) + \mathbb{1}_{\{y \neq k\}} \right) && \text{(Definition)} \\
&\geq \left(f_y(\mathbf{x}) - f_y(\mathbf{x}) + \mathbb{1}_{\{y \neq y\}} \right) && \text{(max always greater than one single component)} \\
&= \mathbb{1}_{\{y \neq y\}} \\
&= 0. && \text{(Indicator function is not true)}
\end{aligned}$$

(b) For sake of convenience, define $g_{k,y}(\mathbf{x}) = f_k(\mathbf{x}) - f_y(\mathbf{x}) + \mathbb{1}_{\{y \neq k\}}$ for any $k, y \in \mathcal{Y}$ and $\mathbf{x} \in \mathcal{X}$. If $k = y$, then of course $g_{k,y}(\mathbf{x}) = 0$, so that in order to find the maximum of the multiclass hinge loss, we need to check for each $k \neq y$ whether $g_{k,y}(\mathbf{x}) > 0$ holds. If this doesn't hold for any $k \neq y$, then the maximum is $0 = g_{y,y}(\mathbf{x})$. Thus,

$$\begin{aligned}
L(y, f(\mathbf{x})) &= \max_k \left(f_k(\mathbf{x}) - f_y(\mathbf{x}) + \mathbb{1}_{\{y \neq k\}} \right) && \text{(Definition)} \\
&= \max_k g_{k,y}(\mathbf{x}) && \text{(Definition of } g_{k,y}(\mathbf{x})) \\
&= \max_{k \neq y} \ \left(\max\{ g_{y,y}(\mathbf{x}), g_{k,y}(\mathbf{x}) \} \right) && \text{(Idea above)} \\
&= \max_{k \neq y} \ \left(\max\{ 0, f_k(\mathbf{x}) - f_y(\mathbf{x}) + \mathbb{1}_{\{y \neq k\}} \} \right) && \text{(Definition of the } g\text{'s)} \\
&= \max_{k \neq y} \ \left(\max\{ 0, f_k(\mathbf{x}) - f_y(\mathbf{x}) + 1 \} \right) && \text{(Indicator function is true)} \\
&\leq \sum_{k \neq y} \max\{ 0, f_k(\mathbf{x}) - f_y(\mathbf{x}) + 1 \}.
\end{aligned}$$

(max is at most the sum over all the **non-negative** components)

(c) In the case of binary classification, i.e., $g = 2$ and $\mathcal{Y} = \{-1, +1\}$, we use a single discriminant model $f(\mathbf{x}) = f_1(\mathbf{x}) - f_{-1}(\mathbf{x})$ based on two scoring functions $f_1, f_{-1} : \mathcal{X} \to \mathbb{R}$ for the prediction by means of $h(\mathbf{x}) = \operatorname{sgn}(f(\mathbf{x}))$. Here, f_1 is the score for the positive class and f_{-1} is the score for the negative class. Show that the upper bound in (b) coincides with the binary hinge loss $L(y, f(\mathbf{x})) = \max\{0, 1 - yf(\mathbf{x})\}$.

We distinguish two cases for $y \in \mathcal{Y} = \{-1, +1\}$.

Case 1: $y = +1$

Then,

$$\begin{aligned}
\sum_{k \neq y} \max\{0, f_k(\mathbf{x}) - f_y(\mathbf{x}) + 1\} &= \sum_{k \neq +1} \max\{0, f_k(\mathbf{x}) - f_1(\mathbf{x}) + 1\} && \text{(Case } y = +1\text{)} \\
&= \max\{0, f_{-1}(\mathbf{x}) - f_1(\mathbf{x}) + 1\} && \text{(Binary classification, i.e., } k \text{ can only be } -1\text{)} \\
&= \max\{0, 1 - f(\mathbf{x})\} && \text{(Definition of } f\text{)} \\
&= \max\{0, 1 - yf(\mathbf{x})\}. && \text{(Case } y = +1\text{)}
\end{aligned}$$

Case 2: $y = -1$

Then,

$$\begin{aligned}
\sum_{k \neq y} \max\{0, f_k(\mathbf{x}) - f_y(\mathbf{x}) + 1\} &= \sum_{k \neq -1} \max\{0, f_k(\mathbf{x}) - f_{-1}(\mathbf{x}) + 1\} && \text{(Case } y = -1\text{)} \\
&= \max\{0, f_1(\mathbf{x}) - f_{-1}(\mathbf{x}) + 1\} && \text{(Binary classification, i.e., } k \text{ can only be } +1\text{)} \\
&= \max\{0, 1 + f(\mathbf{x})\} && \text{(Definition of } f\text{)} \\
&= \max\{0, 1 - yf(\mathbf{x})\}. && \text{(Case } y = -1\text{)}
\end{aligned}$$

(d) Yes, we can state say something similar for the alternative multiclass hinge loss, namely that it is only zero if all the $g - 1$ margins are greater or equal 1, where the margins are $m_{y,k}(\mathbf{x}) = f_y(\mathbf{x}) - f_k(\mathbf{x})$ (y is the true class and $k \in \mathcal{Y} \backslash \{y\}$). Indeed, it holds that

$$\begin{aligned}
m_{y,k}(\mathbf{x}) \geq 1 \quad \forall k \neq y \quad &\Leftrightarrow \quad f_y(\mathbf{x}) - f_k(\mathbf{x}) \geq 1 \quad \forall k \neq y \\
&\Leftrightarrow \quad f_k(\mathbf{x}) - f_y(\mathbf{x}) \leq -1 \quad \forall k \neq y \\
&\Leftrightarrow \quad \max\{0, 1 + f_k(\mathbf{x}) - f_y(\mathbf{x})\} = 0 \quad \forall k \neq y \\
&\Leftrightarrow \quad \sum_{k \neq y} \max\{0, 1 + f_k(\mathbf{x}) - f_y(\mathbf{x})\} = 0
\end{aligned}$$

(e) An empirical loss minimization approach for the (alternative) multiclass hinge loss will try to minimize the $g - 1$ margins *simultaneously*, while the empirical loss minimization approach with the one-vs-rest technique will try to minimize the (binary) margins of the individual binary hinge losses *separately*. In particular, a model obtained by the former will in general not coincide with a model obtained by the latter approach. As an example we can consider the `iris` dataset:

```r
# we use the iris dataset
Z          = data.matrix(iris)
# we add an additional column with all ones for the intercept
Z          = cbind(rep(1,nrow(Z)),Z)
# p is the dimension of the features
p          = ncol(Z)-1
# classes are in the last column of Z
classes    = unique(Z[,p+1])
# g is the number of classes
g          = length(classes)


# one-vs-rest codebook
codebook <-function(y,i){
  if(y==i){
    return (1)
    }
  else{
    return (-1)
    }
}

# binary hinge loss for linear score function (characterized by theta)
bin_hinge <- function(z,theta){
  x = z[1:p]
  y = z[p+1]
  return(max(0,1-y*x%*%t(t(theta))))
}


# empirical risk for binary hinge loss for linear score function (characterized by theta)
emp_risk_bin_hinge <- function(Z,theta){
  sum(apply(Z,1,bin_hinge,theta=theta))
}



# fitting a linear score function in a one-vs-all manner
one_vs_all_bin_theta <-function(Z){
  theta_mat       = matrix(rep(0,p*g),nrow=p)
  for(i in 1:g){
    # recode the last column of the data matrix according to the codebook
    Z_coded       = cbind(Z[,1:p], unlist(sapply(Z[,p+1],codebook,i) ) )
    # finding the best theta
    theta_mat[,i] = optim(rep(0,p),fn=emp_risk_bin_hinge,Z=Z_coded)$par
  }
  return ( theta_mat )
}


# predication with linear score functions in a one-vs-all manner
# hat_theta_mat stores the parameters of the binary classifiers
one_vs_all_bin_predict <- function(hat_theta_mat,x){
  scores = x %*% hat_theta_mat
  return (which(scores==max(scores)))
}

# multiclass hinge loss for linear score functions (characterized by the matrix Theta)
multiclass_hinge<-function(z,Theta){
  Theta  = matrix(Theta,nrow=p,byrow=T)
  x      = z[1:p]
  y      = z[p+1]
  temp   = -1 # we start with -1, as in the for loop below we do not leave out the case k=y
  for(i in 1:g){
    temp = temp + max(0, 1 + Theta[,i]%*%t(t(x)) - Theta[,y]%*%t(t(x))  )
  }
  return(temp)
  # alternative if you do not like for-loops
  #return (sum(pmax(rep(0,g), 1+  t(x)%*%Theta - as.numeric(Theta[,y]%*%t(t(x))) ))-1)
}

# empirical risk for multiclass hinge loss for linear score functions
# characterized by the matrix Theta
```

```
emp_risk_mutliclass_hinge <- function(Z,Theta){
  sum(apply(Z,1,multiclass_hinge,Theta=Theta))
}


# fitting linear score functions for multiclass hinge loss
mutliclass_hinge_theta <-function(Z){
  return ( optim(par=rep(0,p*g),fn=emp_risk_mutliclass_hinge,Z=Z)$par)
}


# predication with linear score functions for multiclass hinge loss
# hat_Theta is the fitted parameter matrix
mutliclass_hinge_predict <-function(hat_Theta,x){
  scores = x %*% hat_Theta
  return (which(scores==max(scores)))
}


# split the iris data into training and test data sets
set.seed(5)
train_ind <- sort(sample(1:nrow(Z),size=50,replace=F))
test_ind <-(1:150)[-train_ind]

# fit the parameters
hat_theta_mat <- one_vs_all_bin_theta(Z[train_ind,])
hat_Theta     <- matrix(mutliclass_hinge_theta(Z[train_ind,]),nrow=p,byrow=T)
hat_Theta

##             [,1]       [,2]       [,3]
## [1,] -0.3501875  0.7707322 -0.3670786
## [2,]  0.1562045  0.3336436 -1.5647621
## [3,]  7.1427582 -2.3870077 -2.9751341
## [4,] -8.1156339  0.1446330  1.4809487
## [5,]  0.2199560  0.1785929  5.3625162

# counting how many times the predictions coincide
count = 0

for(j in test_ind){
  # if predictions differ, print the different predictions and the true label
  if(one_vs_all_bin_predict(hat_theta_mat,Z[j,1:p])!=
     mutliclass_hinge_predict(hat_Theta,Z[j,1:p])){
    print(paste("One_vs_all predicts: ",
                one_vs_all_bin_predict(hat_theta_mat,Z[j,1:p])))
    print(paste("Multiclass hinge loss predicts: ",
                mutliclass_hinge_predict(hat_Theta,Z[j,1:p])))
    print(paste("True label: ",Z[j,p+1]))
    print("")
  }
  else{
    count <- count +1
  }
}

## [1] "One_vs_all predicts:  2"
## [1] "Multiclass hinge loss predicts:  3"
## [1] "True label:  2"
## [1] ""
## [1] "One_vs_all predicts:  3"
## [1] "Multiclass hinge loss predicts:  2"
## [1] "True label:  2"
## [1] ""
## [1] "One_vs_all predicts:  3"
## [1] "Multiclass hinge loss predicts:  2"
## [1] "True label:  2"
## [1] ""
## [1] "One_vs_all predicts:  2"
## [1] "Multiclass hinge loss predicts:  3"
## [1] "True label:  3"
## [1] ""
## [1] "One_vs_all predicts:  2"
## [1] "Multiclass hinge loss predicts:  3"
## [1] "True label:  3"
## [1] ""
```

```
# how many times did the predictions coincide?
count
```

```
## [1] 95
```