ml-1

Prudhvi Peddmallu

24 April 2019

```r
# Required libraries
library(readxl) # reading excel,library(dplyr),library(stringr) # string manipulation,library(kknn) # K
# library(fastICA) # ICA,library(kernlab) # SVM,library(neuralnet) # Neural networks,library(mvtnorm),l
-----------------------------------------------------------
inf = read.csv("given_files/Influenza.csv")
lambdas = seq(10, 2910, 100)
log_poisson = function(lambda, y){
lp = -lambda + y*log(lambda) - sum(log(1:y))
}
calc_llik = function(lambda){
m_llik = 0
for(i in 1:nrow(inf)){
m_llik = m_llik - log_poisson(lambda, inf$Mortality[i])
}
return(m_llik)
}
llik_df = data.frame()
for(lamb in lambdas){
llik_lb = calc_llik(lamb)
llik_df = rbind(llik_df, data.frame(lambda = lamb, minus_log_likelihood = llik_lb))
}
# Plot that shows the dependence of minus log-likelihood on lambda
ggplot(llik_df) + geom_line(aes(x=lambda, y=minus_log_likelihood)) +
ylab("Minus Log likelihood") + xlab("Lambda") + ggtitle("Minus Log likelihood vs Lambda")
inf2 = as.data.frame(scale(inf[ , -3]))
inf2$Mortality = inf$Mortality
n = dim(inf2)[1]
set.seed(12345)
id = sample(1:n, floor(n*0.5))
inf2_train = inf2[id,]
inf2_test = inf2[-id,]
# Perform k-fold CV to select lambda for a LASSO model
inf2_cv_model = cv.glmnet(as.matrix(inf2_train[ , -9]), matrix(inf2_train$Mortality),
alpha = 1, family = "poisson", lambda = seq(0, 30, 0.01))
plot(inf2_cv_model)
# Find optimal lambda and model
optimal_lambda = inf2_cv_model$lambda.min
print(paste("Optimal lambda: ", optimal_lambda))
optimal_lasso_model = glmnet(as.matrix(inf2_train[ , -9]), matrix(inf2_train$Mortality),
alpha = 1, family = "poisson", lambda = optimal_lambda)
pred_trn = predict(optimal_lasso_model, newx = as.matrix(inf2_train[ , -9]), type = "response")
pred_tst = predict(optimal_lasso_model, newx = as.matrix(inf2_test[ , -9]), type = "response")
train_mse = sum((pred_trn - inf2_train$Mortality)^2)/nrow(inf2_train)
test_mse = sum((pred_tst - inf2_test$Mortality)^2)/nrow(inf2_test)
print(paste("Train MSE: ", train_mse))
print(paste("Test MSE: ", test_mse))
print(coef(optimal_lasso_model))
print(paste("alpha = ", coef(optimal_lasso_model)[1]))
print(paste("exp(alpha) = ", exp(coef(optimal_lasso_model)[1])))
inf_9596 = inf[inf$Year == 1995 | inf$Year == 1996, ]
# Benjamini Hochberg method
# Function to compute p-value for given column with Conference
get_p_value = function(col){
form = as.formula(paste(col, "~", "Year"))
res = t.test(form, data = inf_9596, alternative = "two.sided", paired = TRUE)
res$p.value
}
# Compute p-values
p_values = sapply(colnames(inf_9596[ , -1]), get_p_value)
p_value_df = data.frame(feature = colnames(inf_9596[, -1]),
p_value = p_values)
p_value_df = p_value_df[order(p_value_df$p_value), ]
p_value_df$feature_num = 1:nrow(p_value_df)
# Removing Week because the p.value is NaN
p_value_df = p_value_df[p_value_df$feature!="Week", ]
alpha = 0.05
M = nrow(p_value_df)
L = max(which(p_value_df$p_value < (alpha * p_value_df$feature_num / M)))
p_L = p_value_df[L, "p_value"]
# Set hypotheses results
p_value_df$hypo_res = ifelse(p_value_df$p_value <= p_L, "Rejected", "Confirmed")
ggplot(p_value_df) +
geom_point(aes(x = feature_num, y = p_value, color = "Confirmed")) +
geom_abline(slope = alpha/M, intercept = 0) +
labs(color = "Hypothesis Result") +
geom_vline(xintercept = L, linetype = "dashed") +
xlab("Feature number") + ylab("p-value") +
ggtitle("Benjamini-Hochberg plot for hypotheses analysis")
rejected_features = p_value_df$feature[p_value_df$hypo_res == "Rejected"]
pca_inf = prcomp(inf2_train[,-9])
lambda = pca_inf$sdev^2
var_prop = lambda / sum(lambda)
cum_var_prop = cumsum(var_prop)
var_prop_df = data.frame(var_prop = var_prop, cum_var_prop = cum_var_prop,
pc_num = 1:length(var_prop))
kable(var_prop_df)
inf_pc_x = pca_inf$x
infpc_cv_model = cv.glmnet(inf_pc_x, matrix(inf2_train$Mortality),
alpha = 1, family = "poisson", lambda = seq(0, 50, 0.1))
plot(infpc_cv_model)
# Find optimal lambda and model
optimal_pc_lambda = infpc_cv_model$lambda.min
print(paste("Optimal lambda: ", optimal_pc_lambda))
optimal_pc_lasso_model = glmnet(inf_pc_x, matrix(inf2_train$Mortality),
alpha = 1, family = "poisson", lambda = optimal_pc_lambda)
print(coef(optimal_pc_lasso_model))
```