

Praktikum und Übung

Nichtlineare und nichtparametrische Methoden

Sommersemester 2018
S. Döhler

Aufgabe -1. a) Installieren Sie die neuesten Versionen von R , RStudio sowie \LaTeX (MikTeX fuer windows, MacTeX für mac, TexLive fuer Linux).

b) Prüfen Sie die Lauffaehigkeit von R in RStudio indem Sie die R-demos laufen lassen.

Aufgabe 0. a) Studieren Sie sorgfältig die Abschnitte 16.1 und 16.2 aus dem Dokument 'Nonparametric Tests' (entnommen aus 'Introduction to the practice of statistics' von Moore und McCabe) welches Sie in moodle finden. Einen Ausdruck des Dokuments koennen Sie beim Laboringenieur Herrn Schepers erhalten.

b) Wenn Sie sich schon auf den weiteren Verlauf der LV vorbereiten wollen, koennen Sie das Dokument 'Linear Regression' (entnommen aus 'An introduction to statistical learning' von Hastie et al.) auf moodle lesen. Einen Ausdruck des Dokuments koennen Sie beim Laboringenieur Herrn Schepers erhalten.

Table 1:

child	progress	story1	story2
1	high	0.55	0.8
2	high	0.57	0.82
3	high	0.72	0.54
4	high	0.70	0.79
5	high	0.84	0.89
6	low	0.40	0.77
7	low	0.72	0.49
8	low	0.00	0.66
9	low	0.36	0.28
10	low	0.55	0.38

% verfeinerte Zielsetzung

Logging

Die folgenden Logging Daten werden auf den Clients und dem Broker aufgezeichnet:

- Paket ID zur eindeutigen Identifikation
- Timestamp (TS) Package sent und Package received
Jahr-Monat-Tag_Stunden:Minuten:Sekunden:Millisekunden

Geplante Testabläufe

Die Tests werden für die 3 QoS-Modi durchgeführt:

- **0 – once at most** – Nachricht maximal einmal versandt Übertragung hat die gleichen Garantien wie TCP
- **1 – at least once** – Nachricht wird mindestens einmal übertragen

- 2 – exactly once – Nachricht wird genau einmal übertragen

```
setwd("/home/lisa/Darmstadt/05_Speicher und Datennetze IoT/Praktikum/Git/mqtt-qos-roundtrip/logs/latenz-
options(digits.secs=3) # needs to be set from time to time - otherwise R doesn't allow for ms
library("data.table", lib.loc=~R/x86_64-pc-linux-gnu-library/3.4")
library("h2o", lib.loc=~R/x86_64-pc-linux-gnu-library/3.4")
library("tidyr", lib.loc=~R/x86_64-pc-linux-gnu-library/3.4")

#Create the list of log files in the folder
files <- list.files(pattern = "*client1.log", full.names = TRUE, recursive = FALSE)
names <- substr(files, start = 18, stop = 60)

# Read the logs into dataFrames and bind
# df <- rbindlist(lapply(files, fread))

#####
# Create dataFrames #
#####
# Take Date + Time for adequate TS and formate to POSIXct

Timestamp<-c(as.POSIXct("2018-05-18 14:01:41.264 CEST"))
newID<-c()

for (i in 1:length(files)) {
  #x <- get(files[i])
  x<-rbindlist(lapply(files[i], fread))
  colnames(x)<- c("Date", "Time", "Action", "Topic", "QoS", "Size", "ID")

  for (j in 1:nrow(x)) {
    Timestamp[j]<-as.POSIXct(strptime(gsub(":", ".", paste(x[j,1],x[j,2])), "%Y-%m-%d %H.%M.%OS"))
    newID[j]<-paste(x[j,4], x[j,7])
  }
  x<-cbind(x, Timestamp, newID)

  assign(paste(names[i]),x)
}
#> Error in FUN(X[[i]], ...): File './mqtt-roundtrip-qos0-10KByte-1-minutes-100pbs-client1.log' does no

#####
# Create DF to hold RTT #
#####
# name Vector
namesSent<-c()
namesRec<-c()
namesTime<-c()

# Split each set into sent and receive to subtract in next step (each stored separately)
# Create name Vectors for Sent, Receive and Time to access in next step
for (i in 1:length(names)){
  sentTimes <- subset(get(paste(names[i])), Action=="sent")
  recTimes <- subset(get(paste(names[i])), Action=="received")
  assign(paste("sentTimes", names[i]), sentTimes)
  namesSent[i]<-paste("sentTimes", names[i])
  assign(paste("recTimes", names[i]), recTimes)
```

```

namesRec[i]<-paste("recTimes", names[i])

times<-as.data.frame(matrix(nrow=2000, ncol=6)) # Create times Matces to store RTT in next step
colnames(times)<- c("sent", "s_newid", "rec", "r_newid", "rtt", "id")
times[,1] <-as.POSIXct(strptime(times[, "sent"],"%Y-%m-%d %H.%M.%OS"))
times[,3] <-as.POSIXct(strptime(times[, "rec"],"%Y-%m-%d %H.%M.%OS"))
assign(paste("times", names[i]), times)
namesTime[i]<-paste("times", names[i]) # Store Names of Time Matrices to access with get command
}
#> Error in get(paste(names[i])): Objekt 'qos0-10KByte-1-minutes-100pbs-client1.log' nicht gefunden

#####
# Calculate RTT #
#####

# Fill times Data Frames with Sent TS and IDs
for(i in 1 : length(namesSent)){
  sentTimes<- get(paste(namesSent[i]))
  times<- get(paste(namesTime[i]))

  for (j in sentTimes$ID) {
    times[j, "sent"]<- sentTimes[which(sentTimes$ID == j),"Timestamp"]
    times[j, "id"]<- sentTimes[which(sentTimes$ID == j),"ID"]
    times[j, "s_newid"]<- sentTimes[which(sentTimes$ID == j),"newID"]
  }
  assign(paste("times", names[i]), times)
  #assign(times, paste("times", names[i]))
}
#> Error in get(paste(namesSent[i])): ungültiges erstes Argument

# Fill times Data Frames with Recieved TS and IDs
for(i in 1 : length(namesRec)){
  recTimes<- get(paste(namesRec[i]))
  times<- get(paste(namesTime[i]))

  for (j in recTimes$ID) {
    times[j, "rec"]<- recTimes[which(recTimes$ID == j),"Timestamp"]
    times[j, "id"]<- recTimes[which(recTimes$ID == j),"ID"]
    times[j, "r_newid"]<- recTimes[which(recTimes$ID == j),"newID"]
  }
  assign(paste("times", names[i]), times)
  #assign(times, paste("times", names[i]))
}
#> Error in get(paste(namesRec[i])): ungültiges erstes Argument

# Calculate Difference
for (i in 1:length(namesTime)){
  times<- get(paste(namesTime[i]))

  for (j in 1:nrow(times)) {
    times[j,"rtt"]<- difftime(times[j,3], times[j,1])
  }
}

```

```

times <- na.omit(times)
assign(paste("times", names[i]), times)
}
#> Error in get(paste(namesTime[i])): ungültiges erstes Argument

#####
# Merge Data Frames #
#####
latenzTC1mbps <- merge(get(namesTime[1]), get(namesTime[2]), by = "r_newid")
#> Error in get(namesTime[1]): ungültiges erstes Argument

for (i in 1:length(namesTime)){
latenzTC1mbps <- rbind(latenzTC1mbps, get(namesTime[i]))
}
#> Error in rbind(latenzTC1mbps, get(namesTime[i])): Objekt 'latenzTC1mbps' nicht gefunden

#####
# Split Topic Name #
#####
separate(latenzTC1mbps$s_newid)
#> Error in separate(latenzTC1mbps$s_newid): Objekt 'latenzTC1mbps' nicht gefunden

latenzTC1mbpsSep <- latenzTC1mbps %>% separate(s_newid, c("n1", "n2", "QoS", "Size", "Min", "n3", "Speed"))
#> Error in eval(lhs, parent, parent): Objekt 'latenzTC1mbps' nicht gefunden
z <- c(-2, -3, -7, -9)
latenzTC1mbps <- latenzTC1mbpsSep[,z]
#> Error in eval(expr, envir, enclos): Objekt 'latenzTC1mbpsSep' nicht gefunden

#####
# Plot Results #
#####

par(mfrow = c(3, 2))
for (i in 1:length(namesTime)){
time<-get(namesTime[i])
plot(time$id, time$rtt, title(namesTime[1]))
}
#> Error in get(namesTime[i]): ungültiges erstes Argument

```