Online Estimation of Surrogate Measures and its Application to Transportation Safety Analysis

Yilin Zhang

2021/3/23

---

1. Background

1.1. Data Source:

Traffic crashes are rare events, with an average rate of one crash 6.8 per million vehicle miles traveled in the US. The **naturalistic driving study (NDS)** provides an unprecedented opportunity to evaluate crash risk. NDSs are characterized by continuously recording driving information, such as three-dimensional Inertial Measurement Unit (IMU) acceleration, GPS and multi-channel video recordings[1].

- Accurately identifying crashes with robustness
- Benefitting further understanding of driving behavior
- With the overall goal of reducing accidents

This experiment uses the **Second Strategic Highway Research Program (SHRP 2) NDS**, the largest NDS to-date, with more than 3,400 participants and 1 million hours of continuous driving data[2].

1.2 Surrogate Measures:

To mitigate the rarity and insufficiency for traffic crash , a broad spectrum of studies investigate identification and development of various surrogate measures, which helps to supplement the analytical data and enhance the reliability of safety analysis.

To be useful for transportation safety applications, a **surrogate measure** should[4]

- related in a predictable and reliable way to crashes
- converting the non-crash events into a corresponding crash frequency and/or severity
- has physical meaning

Much of the empirical work is framed by detecting the jerky driving behaviors based on the **acceleration information**. The premise is the positive correlation between **elevated gravitational-force events and crash propensity**. While such an attempt is sensitive to the unstable HGF threshold and can easily generate too many false positives.

1.3 Aim of Our Work

Our core concern is to propose surrogates based on three-dimension of acceleration.

1. Propose new surrogate measures with large power in detecting crash
2. Implement the surrogate measures with online version

2. Exploratory Data Analysis

In this section, we do some exploratory data analysis. Our data includes **segments** and **trip** for both crash and baseline scenarios. We use segments as training data and trip data as testing.

- Training Data: Each driving segment is about 200 time points, 1000 baseline segments and 400 crash segments.
- Testing Data: Each driving trip is about 2000 time points, 500 baseline trips and 200 crash trips.

```
library(plyr)
library(dplyr)
library(ggplot2)
library(reshape2)
library(e1071)
library(ggridges)
library(gridExtra)
library(pROC)
library(mgcv)
library(gbm)
```

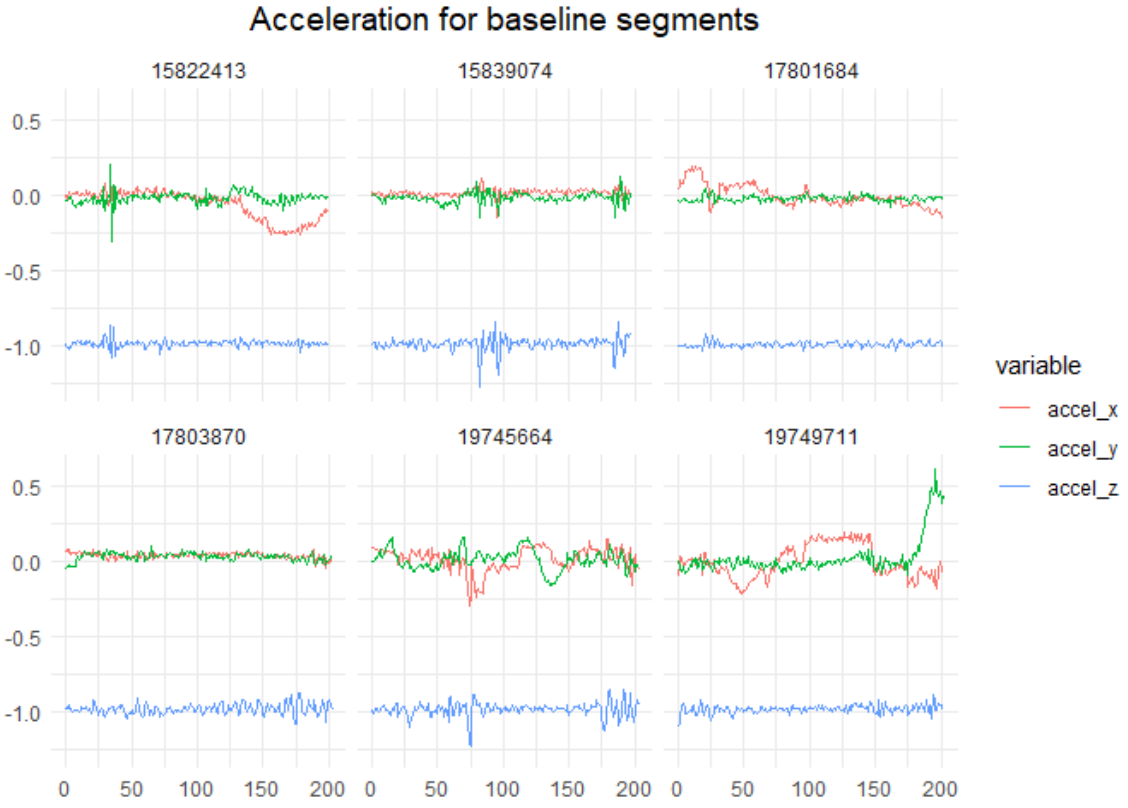
```
dir_crash <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/segment/crash/"
setwd(dir_crash)
f_crash <- list.files(dir_crash)
data_crash <- ldply(f_crash[1:6], read.csv, header=TRUE)
data_crash <- mutate(data_crash, crash = 1)


dir_base <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/segment/base/"
setwd(dir_base)
f_base <- list.files(dir_base)
data_base <- ldply(f_base[1:6], read.csv, header=TRUE)
data_base <- mutate(data_base, crash = 0)

dataset <- bind_rows(data_base, data_crash)
```

2.1. Basic description for driving segments

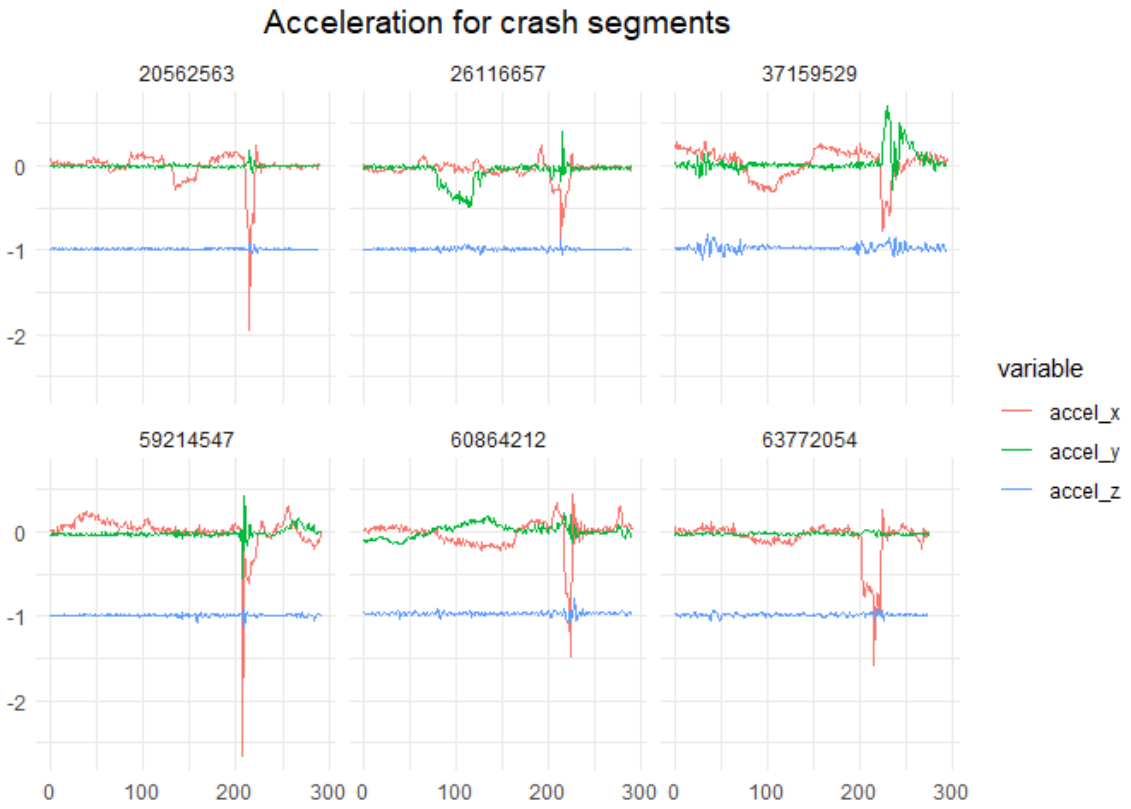
For general understanding of the segments, we plot 6 segments for both crash and baseline ones. From the two figures below, we can see that,

- The accelerations in three dimensions for baseline segments are more stationary than the crash ones.
- The accelerations in three dimensions for crash ones appear different patterns. This indicates the various cause of driving behaviors for crash.

```r
# figure for baseline segments
temp <- data_base %>% select(file_id, X, accel_x, accel_y, accel_z)
temp <- melt(temp, id.vars = c("file_id", "X"))


p <- temp %>%
  ggplot(aes(x=X, y=value, group=variable, color=variable)) +
  geom_line(size=0.7) +
  facet_wrap("file_id")+
  ggtitle("Acceleration for baseline segments") + theme_minimal()+
  theme(plot.title = element_text(size = 14, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())
p
```

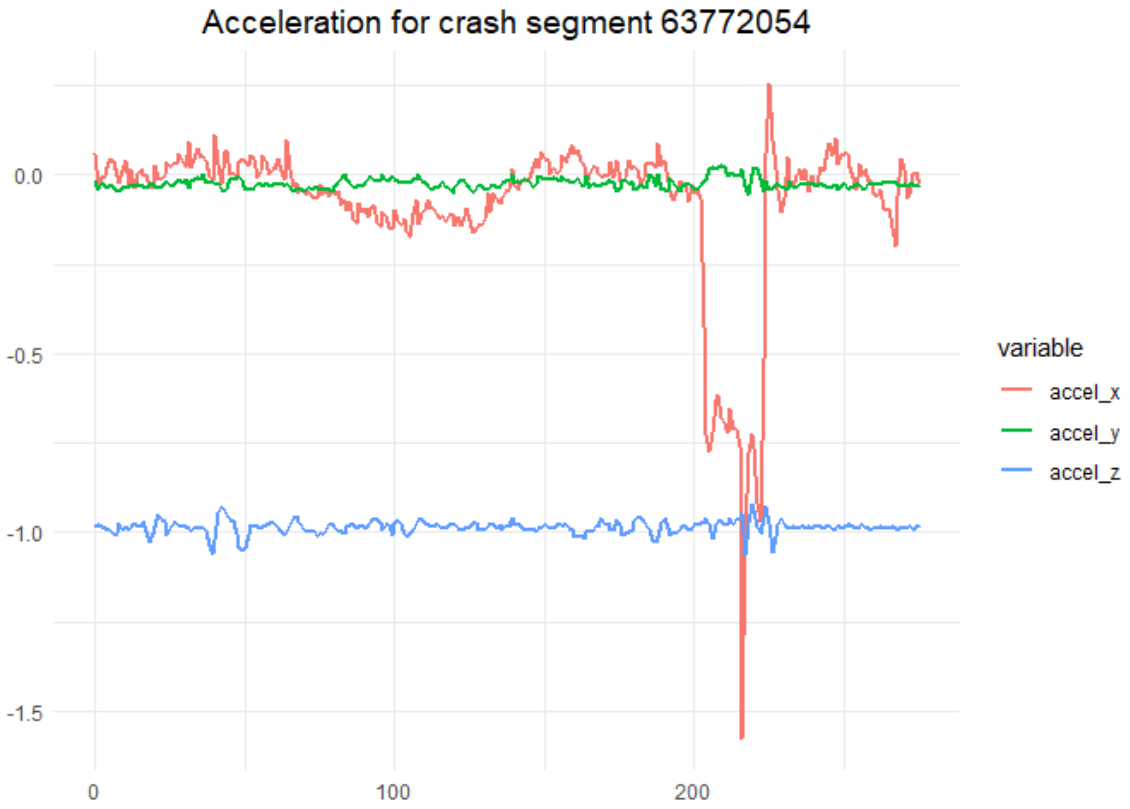



```r
temp <- data_crash %>% select(file_id, X, accel_x, accel_y, accel_z)
temp <- melt(temp, id.vars = c("file_id", "X"))

p <- temp %>%
  ggplot(aes(x=X, y=value, group=variable, color=variable)) +
    geom_line(size=0.7) +
    facet_wrap("file_id")+
    ggtitle("Acceleration for crash segments") + theme_minimal()+
    theme(plot.title = element_text(size = 14, hjust = 0.5),
          axis.title.x=element_blank(),
          axis.title.y=element_blank())
p
```



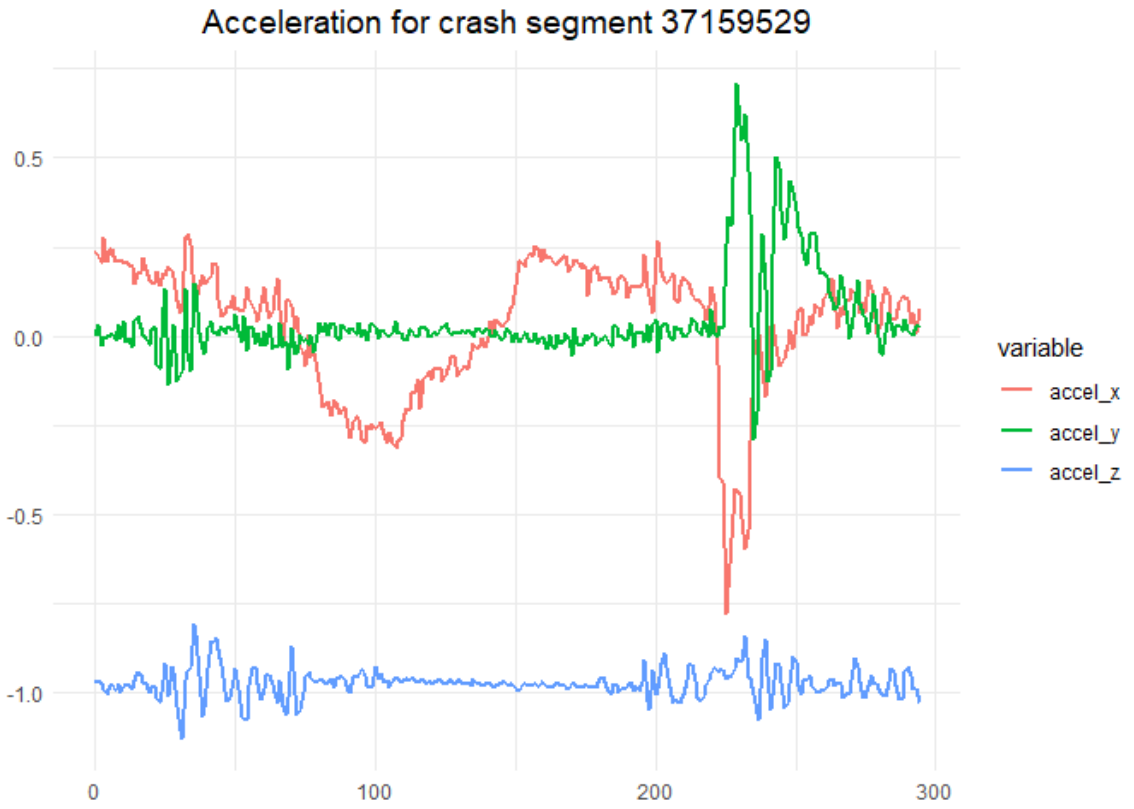
For further analyze the driving behaviors for crash, we plot two figures for crash segments.

```
# figure for crash segment 63772054
temp <- data_crash %>%
  filter(file_id == "63772054") %>%
  select(X, accel_x, accel_y, accel_z)
temp <- melt(temp, id.vars = c("X"))


p <- temp %>%
  ggplot(aes(x=X, y=value, group=variable, color=variable)) +
  geom_line(size=1.0) +
  ggtitle("Acceleration for crash segment 63772054") + theme_minimal() +
  theme(plot.title = element_text(size = 14, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())
p
```



The above figure depicts that The driver slamming on the brakes and crash into the object in the front of it.

```
# figure for crash segment 37159529
temp <- data_crash %>%
  filter(file_id == "37159529") %>%
  select(X, accel_x, accel_y, accel_z)
temp <- melt(temp, id.vars = c("X"))


p <- temp %>%
  ggplot(aes(x=X, y=value, group=variable, color=variable)) +
  geom_line(size=1.0) +
  ggtitle("Acceleration for crash segment 37159529") + theme_minimal() +
  theme(plot.title = element_text(size = 14, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())
p
```



The above figure depicts that the driver slamming on the brakes and hitting the steering wheel.

2.2. Crash V.S. baseline density

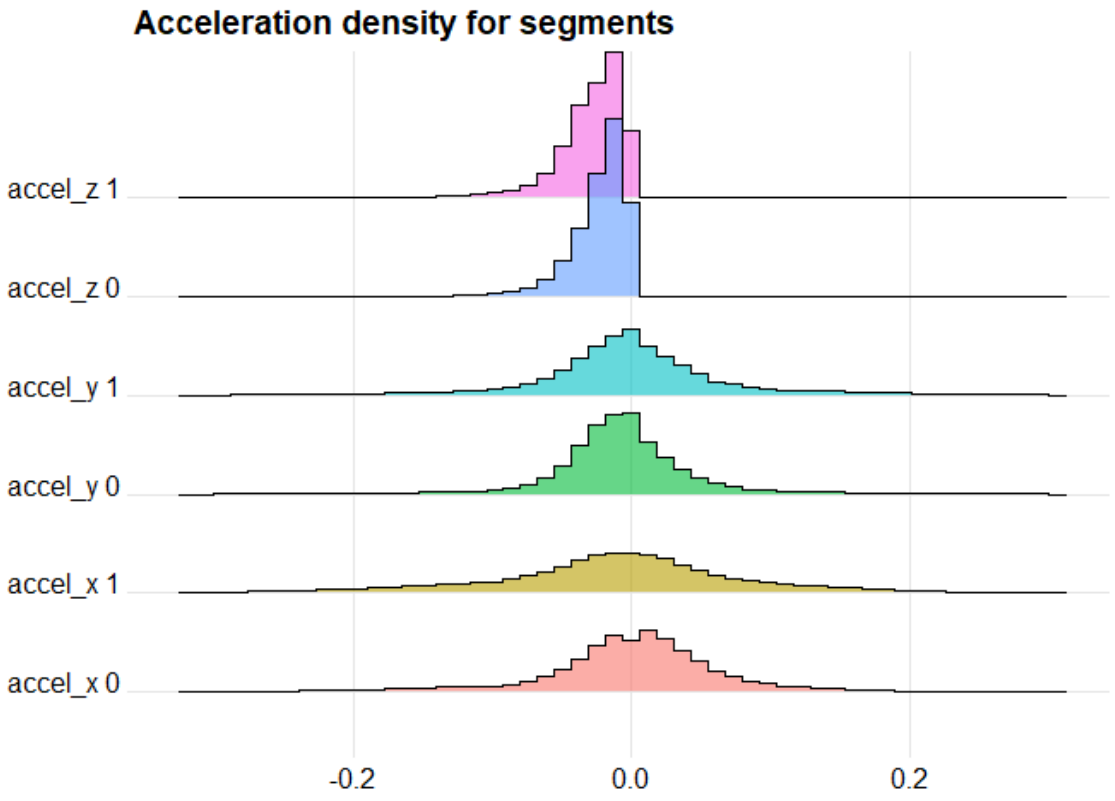
To further explore the difference between crash and baseline samples, we plot the density for different two cases.

```
# import all segment data
dir_crash <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/segment/crash/"
setwd(dir_crash)
f_crash <- list.files(dir_crash)
data_crash <- ldply(f_crash, read.csv, header=TRUE)
data_crash <- mutate(data_crash, crash = 1)


dir_base <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/segment/base/"
setwd(dir_base)
f_base <- list.files(dir_base)
data_base <- ldply(f_base, read.csv, header=TRUE)
data_base <- mutate(data_base, crash = 0)

dataset <- bind_rows(data_base, data_crash)
```

```
# import all trip data
dir_crash <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/trip/crash/"
setwd(dir_crash)
f_crash <- list.files(dir_crash)
trip_crash <- ldply(f_crash, read.csv, header=TRUE)
```

```
trip_crash <- mutate(trip_crash, crash = 1)


dir_base <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/trip/base/"
setwd(dir_base)
f_base <- list.files(dir_base)
trip_base <- ldply(f_base, read.csv, header=TRUE)
trip_base <- mutate(trip_base, crash = 0)

tripset <- bind_rows(trip_base, trip_crash)
```

```
temp <- dataset %>%
  select(crash, accel_x, accel_y, accel_z)
log_d1 <- function(x){
  return(log(abs(x)+1)*sign(x))
}
log_d2 <- function(x){
  return(log(abs(x+1)+1)*sign(x))
}
temp["accel_x"] <- lapply(temp["accel_x"], log_d1)
temp["accel_y"] <- lapply(temp["accel_y"], log_d1)
temp["accel_z"] <- lapply(temp["accel_z"], log_d2)
temp <- melt(temp, id.vars = c("crash"))
temp$label <- paste(temp$variable, temp$crash)

temp <- temp %>%
  filter(value<0.3) %>% filter(value>-0.3)

p <- temp %>%
  ggplot( aes(y=label, x=value,  fill=label)) +
  geom_density_ridges(alpha=0.6, stat="binline", bins=50) +
  theme_ridges() +
  ggtitle("Acceleration density for segments") +
  theme(
    legend.position="none",
    panel.spacing = unit(0.1, "lines"),
    strip.text.x = element_text(size = 1),
    axis.title.x=element_blank(),
    axis.title.y=element_blank())
p
```



The above density of acceleration for crash and non-crash data in three dimension and find

- the rare event account for a small part but with extreme value
- hard to distinguish crash and non-crash only from extreme value

This highly motivates us to propose new surrogates.

2.3 New surrogates

We are going to propose new surrogate measures that

- has large power in detecting crash
- has low false positives

We propose three new surrogates to distinguish crash and baseline ones, include

- standard deviation
- coefficient of variation
- skewness

For each segments, we calculate these surrogates and depict the violin plots for these surrogates, compared with the previous proposed one.

- maximum

1. The new surrogate **standard deviation**.

```
# figure for standard deviation
temp <- dataset %>%
  select(crash, file_id, accel_x, accel_y, accel_z) %>%
```

```
  group_by(crash, file_id) %>%
  summarise(
    ac_std_x = sd(accel_x),
    ac_std_y = sd(accel_y),
    ac_std_z = sd(accel_z)
  )%>%
  select(crash, ac_std_x, ac_std_y, ac_std_z) %>%
  melt(id.vars = c("crash"))
temp$crash <- factor(temp$crash)

p1 <- ggplot(temp, aes(x=variable, y=value, fill=crash)) +
  geom_violin()+
  ggtitle("Standard deviation for segments") + theme_minimal()+
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())
```

2. The new surrogate **Coefficient of variation**.

```
temp <- dataset %>%
  select(crash, file_id, accel_x, accel_y, accel_z) %>%
  group_by(crash, file_id) %>%
  summarise(
    ac_cv_x = sd(accel_x)/(mean(accel_x)),
    ac_cv_y = sd(accel_y)/(mean(accel_y)),
    ac_cv_z = sd(accel_z)/(mean(accel_z))
  )%>%
  select(crash, ac_cv_x, ac_cv_y, ac_cv_z) %>%
  filter(ac_cv_x>-25) %>%
  filter(ac_cv_x<25) %>%
  filter(ac_cv_y>-25) %>%
  filter(ac_cv_y<25) %>%
  melt(id.vars = c("crash"))
temp$crash <- factor(temp$crash)

p2 <- ggplot(temp, aes(x=variable, y=value, fill=crash)) +
  geom_boxplot()+
  ggtitle("Coefficient of variation for segments") + theme_minimal()+
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())
```

3. The a new surrogate **skewness**.

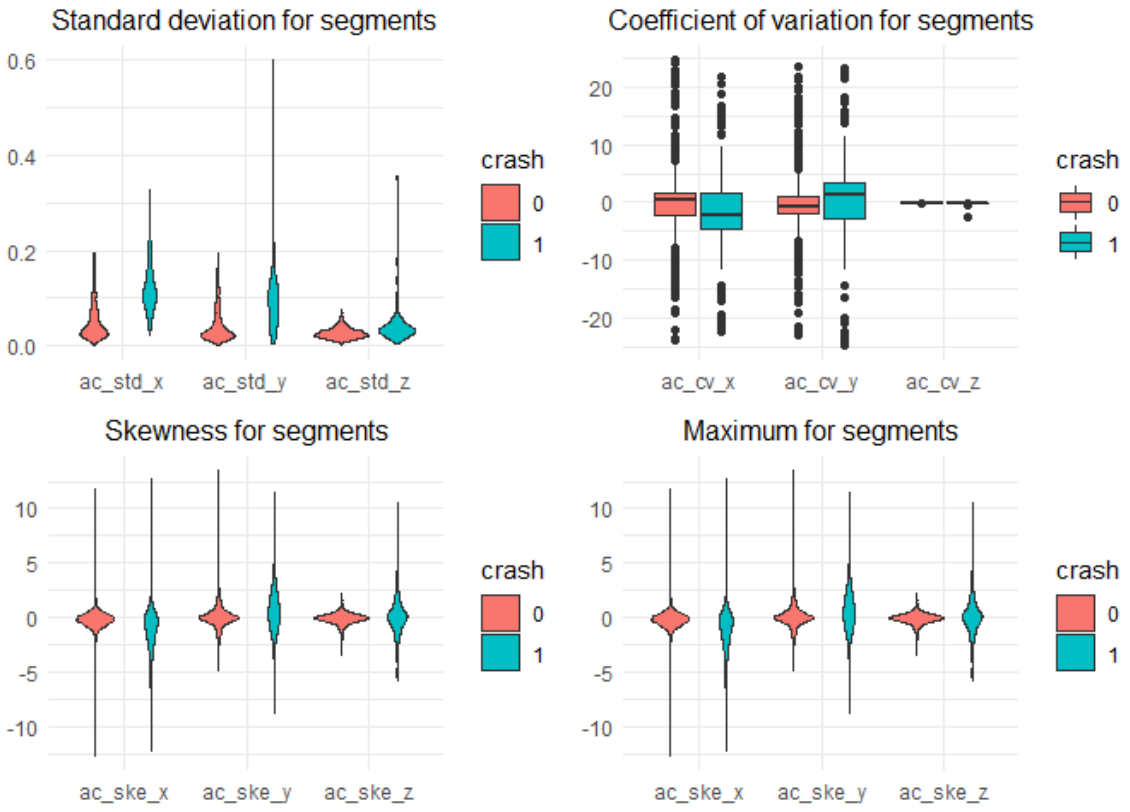
```
temp <- dataset %>%
  select(crash, file_id, accel_x, accel_y, accel_z) %>%
  group_by(crash, file_id) %>%
  summarise(
    ac_ske_x = skewness(accel_x, type = 1),
    ac_ske_y = skewness(accel_y, type = 1),
```

```
    ac_ske_z = skewness(accel_z, type = 1)
  )%>%
  select(crash, ac_ske_x, ac_ske_y, ac_ske_z) %>%
  melt(id.vars = c("crash"))
temp$crash <- factor(temp$crash)

p3 <- ggplot(temp, aes(x=variable, y=value, fill=crash)) +
  geom_violin()+
  ggtitle("Skewness for segments") + theme_minimal() +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())
```

4. The surrogate **Maximum**, proposed previously.

```
temp <- dataset %>%
  select(crash, file_id, accel_x, accel_y, accel_z) %>%
  group_by(crash, file_id) %>%
  summarise(
    ac_ske_x = skewness(accel_x, type = 1),
    ac_ske_y = skewness(accel_y, type = 1),
    ac_ske_z = skewness(accel_z, type = 1)
  )%>%
  select(crash, ac_ske_x, ac_ske_y, ac_ske_z) %>%
  melt(id.vars = c("crash"))
temp$crash <- factor(temp$crash)

p4 <- ggplot(temp, aes(x=variable, y=value, fill=crash)) +
  geom_violin()+
  ggtitle("Maximum for segments") + theme_minimal() +
  theme(plot.title = element_text(size = 12, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())
```

```
l <- list(p1, p2, p3, p4)
grid.arrange(grobs = l, ncol = 2)
```



From the figures above, we can see that

- standard deviation or skewness could be better surrogates
- the surrogates on dimensional z may be useless.

Therefore, we choose our surrogates as **standard deviation**, **variation of coefficient** and **skewness**. For each surrogate, we consider **x** and **y** dimension. Totoally, we have 6 features in prediction model.

3. Formal model

We choose four different models to predict the risk for crash.

- Logistic regression
- Generalized addictive model
- Support vactor machine
- Gradient boosting decision tree

For each prediction model, we consider two sub models with different surrogates.

- one is three **new surrogates** (standard deviation, coefficient of variation, skewness) based on dimension x and y
- one is surrogates proposed previously (maximum) based on dimension x and y

The training data is segment data and the testing is trip one.

```
# acquire train and test data
train_data <- dataset %>%
  group_by(crash, file_id) %>%
  summarise(
```

```
    ac_std_x = sd(accel_x),
    ac_std_y = sd(accel_y),
    ac_cv_x = sd(accel_x)/(mean(accel_x)),
    ac_cv_y = sd(accel_y)/(mean(accel_y)),
    ac_ske_x = skewness(accel_x, type = 1),
    ac_ske_y = skewness(accel_y, type = 1),
    ac_max_x = max(accel_x),
    ac_max_y = max(accel_y)
  )


test_data <- tripset %>%
  group_by(crash, file_id) %>%
  summarise(
    ac_std_x = sd(accel_x),
    ac_std_y = sd(accel_y),
    ac_cv_x = sd(accel_x)/(mean(accel_x)),
    ac_cv_y = sd(accel_y)/(mean(accel_y)),
    ac_ske_x = skewness(accel_x, type = 1),
    ac_ske_y = skewness(accel_y, type = 1),
    ac_max_x = max(accel_x),
    ac_max_y = max(accel_y)
  )

train_data$crash <- factor(train_data$crash)
```

1. Logistic Model

```
# logistic regression
logit.sur <- glm(crash ~ ac_std_x + ac_std_y + ac_cv_x + ac_cv_y+ ac_ske_x + ac_ske_y,
                 data = train_data, family = "binomial")
logit.max <- glm(crash ~ ac_max_x + ac_max_y,
                 data = train_data, family = "binomial")
test_data$p.sur <- predict(logit.sur, newdata = test_data, type = "response")
test_data$p.max <- predict(logit.max, newdata = test_data, type = "response")


roc.logit.sur <- roc(test_data$crash, test_data$p.sur)
roc.logit.max <- roc(test_data$crash, test_data$p.max)
roclist <- list("Ours" = roc.logit.sur, "Max" = roc.logit.max)
g.logit <- ggroc(roclist, aes = "linetype", legacy.axes = TRUE) +
  geom_abline() +
  theme_classic() +
  ggtitle("AOC curve for Logistic regression") +
  theme(plot.title = element_text(size = 10, hjust = 0.5)) +
  labs(x = "1-Specificity",
       y = "Sensitivity")
```

2. Generalized addictive model

```r
gam.sur <- gam(crash ~ s(ac_std_x) + s(ac_std_y) + s(ac_cv_x) +
                 s(ac_cv_y) + s(ac_ske_x) + s(ac_ske_y),
                 data = train_data, family = binomial())

gam.max <- gam(crash ~ s(ac_max_x) + s(ac_max_y),
                 data = train_data, family = binomial())

test_data$p.sur <- predict(gam.sur, newdata = test_data, type = "response")
test_data$p.max <- predict(gam.max, newdata = test_data, type = "response")

roc.gam.sur = roc(test_data$crash, test_data$p.sur)
roc.gam.max = roc(test_data$crash, test_data$p.max)

roclist <- list("Ours" = roc.gam.sur, "Max" = roc.gam.max)
g.gam <- ggroc(roclist, aes = "linetype", legacy.axes = TRUE) +
  geom_abline() +
  theme_classic() +
  ggtitle("AOC curve for generilzed addictive model") +
  theme(plot.title = element_text(size = 10, hjust = 0.5)) +
  labs(x = "1-Specificity",
       y = "Sensitivity")
```

3. Support vector machine

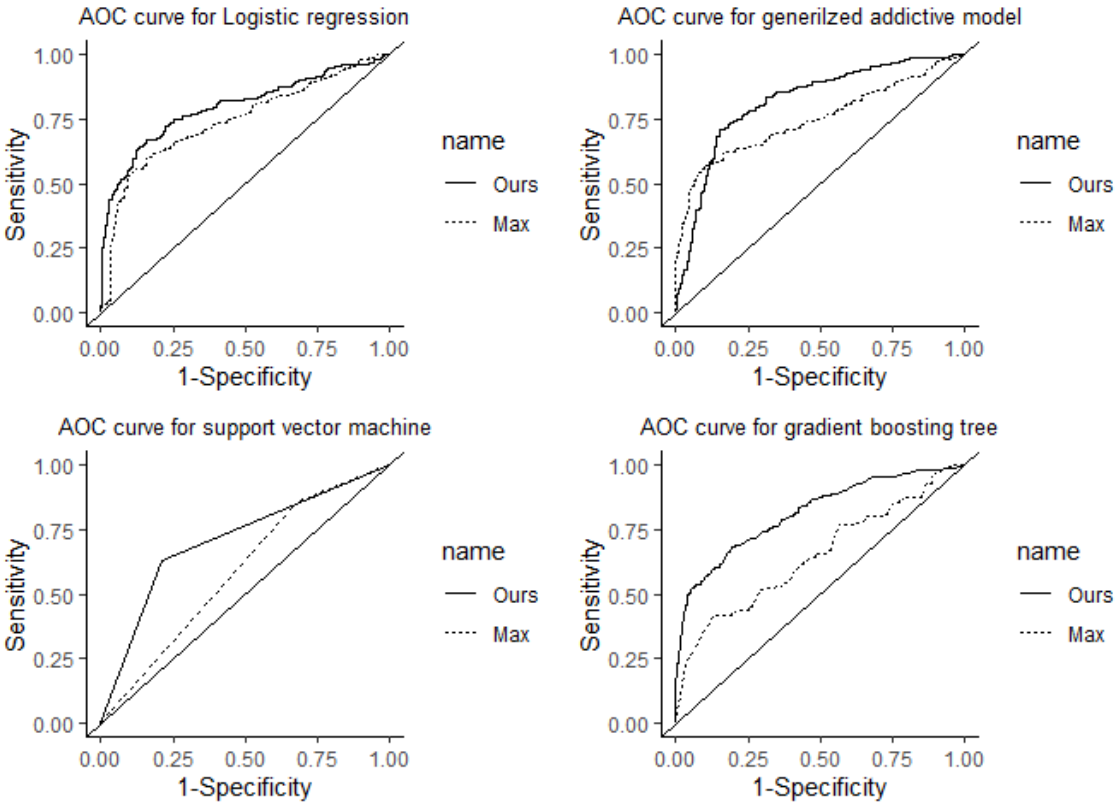
```r
temp_train_sur <- train_data %>% dplyr::select(crash, ac_std_x, ac_std_y, ac_cv_x, ac_cv_y,
                                    ac_ske_x, ac_ske_y)
temp_train_max <- train_data %>% dplyr::select(crash, ac_max_x, ac_max_y)
temp_test_sur <- test_data %>% dplyr::select(crash, ac_std_x, ac_std_y, ac_cv_x, ac_cv_y,
                                          ac_ske_x, ac_ske_y)
temp_test_max <- test_data %>% dplyr::select(crash, ac_max_x, ac_max_y)


svm.sur <- svm(crash ~ ., data = temp_train_sur,
               kernel = "radial", cost = 20)
svm.max <- svm(crash ~ ., data = temp_train_max,
               kernel = "radial", cost = 20)


test_data$p.sur <- as.integer(predict(svm.sur, newdata = temp_test_sur))-1
test_data$p.max <- as.integer(predict(svm.max, newdata = temp_test_max))-1

roc.svm.sur <- roc(test_data$crash, test_data$p.sur)
roc.svm.max <- roc(test_data$crash, test_data$p.max)

roclist <- list("Ours" = roc.svm.sur, "Max" = roc.svm.max)
g.svm <- ggroc(roclist, aes = "linetype", legacy.axes = TRUE) +
  geom_abline() +
  theme_classic() +
  ggtitle("AOC curve for support vector machine") +
  theme(plot.title = element_text(size = 10, hjust = 0.5)) +
```

```
  labs(x = "1-Specificity",
       y = "Sensitivity")
```

4. Gradient boosting decision tree

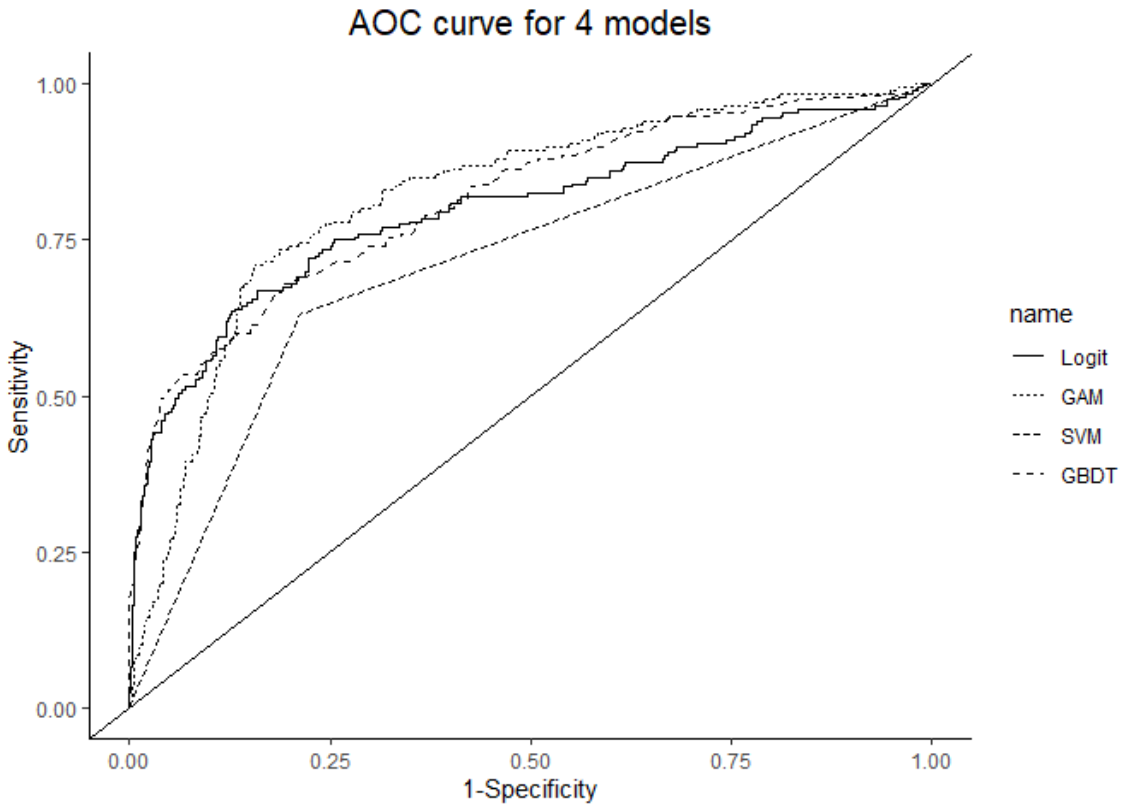
```
temp_train_sur <- train_data %>% dplyr::select(crash, ac_std_x, ac_std_y, ac_cv_x, ac_cv_y,
                                               ac_ske_x, ac_ske_y)
temp_train_max <- train_data %>% dplyr::select(crash, ac_max_x, ac_max_y)
temp_test_sur <- test_data %>% dplyr::select(crash, ac_std_x, ac_std_y, ac_cv_x, ac_cv_y,
                                             ac_ske_x, ac_ske_y)
temp_test_max <- test_data %>% dplyr::select(crash, ac_max_x, ac_max_y)
temp_train_sur <- na.omit(temp_train_sur)
temp_train_sur$crash <- as.integer(temp_train_sur$crash)-1
temp_train_max <- na.omit(temp_train_max)
temp_train_max$crash <- as.integer(temp_train_max$crash)-1

gbdt.sur <- gbm(crash ~ ., data = temp_train_sur,
                shrinkage=0.01, distribution = 'bernoulli')
gbdt.max <- gbm(crash ~ ., data = temp_train_max,
                shrinkage=0.01, distribution = 'bernoulli')


test_data$p.sur <- predict(gbdt.sur, temp_test_sur, na.action = na.pass)
test_data$p.max <- predict(gbdt.max, newdata = temp_test_max)

roc.gbdt.sur <- roc(test_data$crash, test_data$p.sur)
roc.gbdt.max <- roc(test_data$crash, test_data$p.max)

roclist <- list("Ours" = roc.gbdt.sur, "Max" = roc.gbdt.max)
g.gbdt <- ggroc(roclist, aes = "linetype", legacy.axes = TRUE) +
  geom_abline() +
  theme_classic() +
  theme(plot.title = element_text(size = 10, hjust = 0.5)) +
  ggtitle("AOC curve for gradient boosting tree") +
  labs(x = "1-Specificity",
       y = "Sensitivity")
```

```
# all model for risk prediction
grid.arrange(g.logit, g.gam, g.svm, g.gbdt, ncol = 2)
```



We plot AOC curves for these 4 risk prediction models each with two curves of our surrogates and maximum surrogate. From the picture above, we can see that to attain high sensitivity, our surrogates works better that the maximum in all four models.

```
# AOC curve for our surrogates with different models
roclist <- list("Logit" = roc.logit.sur, "GAM" = roc.gam.sur,
                "SVM" = roc.svm.sur, "GBDT" = roc.gbdt.sur)
g.compare <- ggroc(roclist, aes = "linetype", legacy.axes = TRUE) +
  geom_abline() +
  theme_classic() +
  ggtitle("AOC curve for 4 models") +
  theme(plot.title = element_text(size = 15, hjust = 0.5)) +
  labs(x = "1-Specificity",
       y = "Sensitivity")
g.compare
```



We then plot AOC curves For 4 prediction models with our surrogates. From the above figure, we know that the generalized addictive model has best performance.

To this end, we construct the risk prediction model use our new surrogates via generalized addictive model (GAM).

4. Online surrogates and risk prediction

4.1 Implement surrogates with online version

Considering that all the surrogates we proposed, can be estimated using U statistics or function of U statistics. Thus, we derive a general frame work for online U statistics.

- In the first step, we calculate the U statistics based on all initial data, which is \[\hat\theta^{(0)} = (C_{n}^{m})^{-1}\sum_{ \{i_1,\ldots,i_m\}\in I_0} h(X_{i_1},\ldots,h_{i_m}).\]
- To refresh the U statistics in time \(t\), we use the following iterative formula.

\[R^{(t)} = \sum\limits_{\{i_1,\ldots,i_m\}\in I_t}h(X_{i_1},\ldots,X_{i_m}) , \]

\[C^{(t)} = \sum_{k=1}^{m-1}\sum\limits_{\{i_1,\ldots,i_k\}\in J_{t-1}, \{i_{k+1},\ldots,i_m\}\in I_{t}}h(X_{i_1},\ldots,X_{i_m})\]

\[\hat\theta^{(t)} = \left\{(t+1)C_n^m-tC_r^m\right\}^{-1}\left[\left\{tC_n^m-(t-1)C_r^m\right\}\hat\theta^{(t-1)} + C^{(t)} + R^{(t)}\right]\]

- By transforming \(\hat\theta^{(t)}\) as \(T(\hat\theta^{(t)})\), we derive the new estimates for \(T(\theta)\).

4.2 Onlne surrogates for crash sample

To examine the performance of online estimation, we depict the longitudinal and lateral online estimation for our three new surrogates. These figures indicates that our online estimation capture the features as time goes by.

```
online_std <- function(x, r, n){
  N <- length(x)
  u <- c()
  ss <- n
  t <- 0

  u_r <- max(abs(x[1:n]))
  u <- c(u, u_r)

  while((ss+r)<=length(x)){

    t <- t + 1

    R <- (r*(r-1)/2)*var(x[(ss+1):(ss+r)])
    C <- sum(outer(x[(ss-n+1+r):ss], x[(ss+1):(ss+r)], FUN = "-")^2)/2

    u_r <- u_r*(n*(n-1)/2 + (t-1)*r*(r-1)/2 + (t-1)*(n-r)*r)
    u_r <- u_r + R + C
    u_r <- u_r/(n*(n-1)/2 + t*r*(r-1)/2 + t*(n-r)*r)

    u <- c(u, u_r)
    ss <- ss + r
  }
  return(sqrt(u))
}
```

```
online_cv <- function(x, r, n){
  N <- length(x)
  u1 <- c()
  u2 <- c()
  ss <- n
  t <- 0

  u1_r <- var(x[1:n])
  u1 <- c(u1, u1_r)

  u2_r <- mean(x[1:n])
  u2 <- c(u2, u2_r)


  while((ss+r)<=length(x)){

    t <- t + 1
```

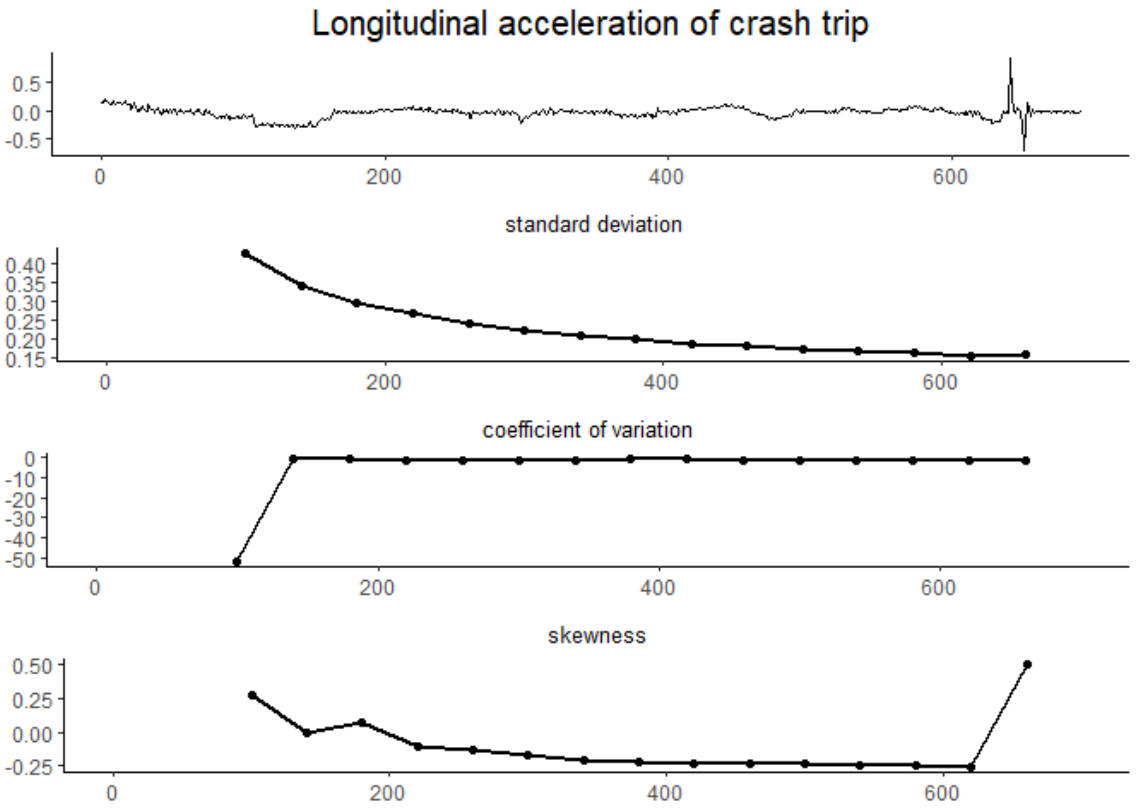
```
    R1 <- (r*(r-1)/2)*var(x[(ss+1):(ss+r)])
    C1 <- sum(outer(x[(ss-n+1+r):ss], x[(ss+1):(ss+r)], FUN = "-")^2)/2

    u1_r <- u1_r*(n*(n-1)/2 + (t-1)*r*(r-1)/2 + (t-1)*(n-r)*r)
    u1_r <- u1_r + R1 + C1
    u1_r <- u1_r/(n*(n-1)/2 + t*r*(r-1)/2 + t*(n-r)*r)

    u1 <- c(u1, u1_r)


    R2 <- sum(x[(ss+1):(ss+r)])
    u2_r <- (u2_r*(n + (t-1)*r) + R2)/(n + t*r)

    u2 <- c(u2, u2_r)


    ss <- ss + r
  }
  return(sqrt(u1)/u2)
}
```

```
online_skew <- function(x, r, n){
  N <- length(x)
  u1 <- c()
  u2 <- c()
  ss <- n
  t <- 0


  u1_r <-  n*1.0/((n-1)*(n-2))*sum((x[1:ss] - mean(x[1:ss]))^3)
  u1 <- c(u1, u1_r)

  u2_r <- var(x[1:n])
  u2 <- c(u2, u2_r)

  while((ss+r)<=length(x)){

    t <- t + 1

    R1 <- r*r/6.0*sum((x[(ss+1):(ss+r)] - mean(x[(ss+1):(ss+r)]))^3)
    temp <- (n-r)*(n-r)/6.0*sum((x[(ss-n+r+1):ss] - mean(x[(ss-n+r+1):ss]))^3)
    C1 <- n*n/6.0*sum((x[(ss-n+r+1):(ss+r)] - mean(x[(ss-n+r+1):(ss+r)]))**3)
           - temp - R1

    u1_r <- u1_r*(n*(n-1)*(n-2)/6.0 + (t-1)*r*(r-1)*(r-2)/6.0
                  + (t-1)*(n-r)*(n-r-1)*r/2.0 + (t-1)*(n-r)*r*(r-1)/2.0)
    u1_r <- u1_r + R1 + C1
    u1_r <- u1_r/(n*(n-1)*(n-2)/6.0 + t*r*(r-1)*(r-2)/6.0
                  + t*(n-r)*(n-r-1)*r/2.0 + t*(n-r)*r*(r-1)/2.0)

    u1 <- c(u1, u1_r)
```

```
    R2 <- (r*(r-1)/2)*var(x[(ss+1):(ss+r)])
    C2 <- sum(outer(x[(ss-n+1+r):ss], x[(ss+1):(ss+r)], FUN = "-")^2)/2

    u2_r <- u2_r*(n*(n-1)/2 + (t-1)*r*(r-1)/2 + (t-1)*(n-r)*r)
    u2_r <- u2_r + R2 + C2
    u2_r <- u2_r/(n*(n-1)/2 + t*r*(r-1)/2 + t*(n-r)*r)

    u2 <- c(u2, u2_r)
    ss <- ss + r
  }
  return(u1/u2^1.5)
}
```

```
dir_crash <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/trip/crash/"
setwd(dir_crash)
f_crash <- list.files(dir_crash)
trip_crash <- ldply(f_crash[20], read.csv, header=TRUE)


x_std <- online_std(trip_crash$accel_x, 40, 100)
x_cv <- online_cv(trip_crash$accel_x, 40, 100)
x_skew <- online_skew(trip_crash$accel_x, 40, 100)
t <- c(0:(length(x_std)-1))*40+100
temp <- data.frame(t, x_std, x_cv, x_skew)

p1 <- trip_crash %>%
  ggplot(aes(x=t, y=accel_x)) +
  geom_line(size=0.7) +
  ggtitle("Longitudinal acceleration of crash trip") + theme_classic()+
  theme(plot.title = element_text(size = 15, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())


p2 <- temp %>%
  ggplot(aes(x=t, y=x_std)) +
  geom_line(size=1.0) + geom_point() +
  ggtitle("standard deviation") + theme_classic() +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank()) +
  scale_x_continuous(limits=c(0, 700))

p3 <- temp %>%
  ggplot(aes(x=t, y=x_cv)) +
  geom_line(size=1.0) + geom_point()+
  ggtitle("coefficient of variation") + theme_classic()+
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
```

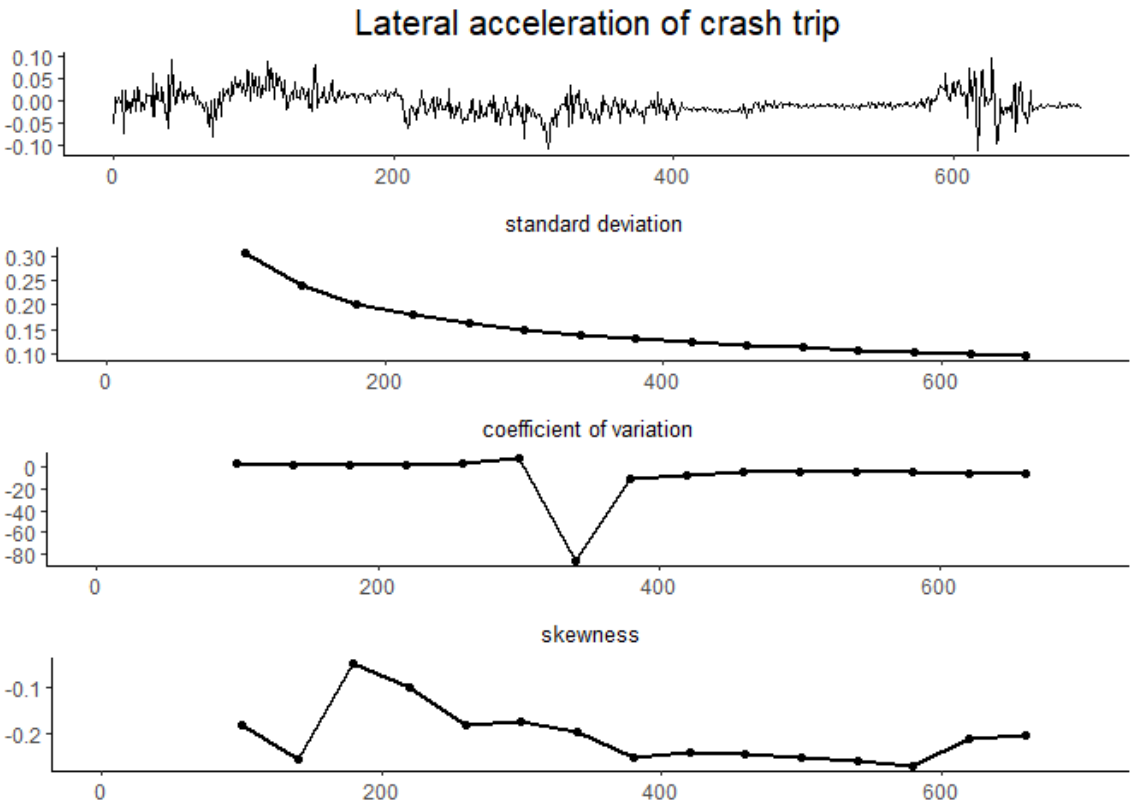
```
        axis.title.y=element_blank()) +
  scale_x_continuous(limits=c(0, 700))

p4 <- temp %>%
  ggplot(aes(x=t, y=x_skew)) +
  geom_line(size=1.0) + geom_point() +
  ggtitle("skewness") + theme_classic() +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank()) +
  scale_x_continuous(limits=c(0, 700))

grid.arrange(p1, p2, p3, p4, ncol = 1)
```



The above figure shows the longitudinal acceleration of crash trip and online estimation for our proposed surrogates. From the figure, we find that our online estimates changes as accelerations change.

```
y_std <- online_std(trip_crash$accel_y, 40, 100)
y_cv <- online_cv(trip_crash$accel_y, 40, 100)
y_skew <- online_skew(trip_crash$accel_y, 40, 100)
t <- c(0:(length(x_std)-1))*40+100
temp <- data.frame(t, y_std, y_cv, y_skew)

p1 <- trip_crash %>%
  ggplot(aes(x=t, y=accel_y)) +
  geom_line(size=0.7) +
  ggtitle("Lateral acceleration of crash trip") + theme_classic()+
  theme(plot.title = element_text(size = 15, hjust = 0.5),
```

```
        axis.title.x=element_blank(),
        axis.title.y=element_blank())


p2 <- temp %>%
  ggplot(aes(x=t, y=y_std)) +
  geom_line(size=1.0) + geom_point() +
  ggtitle("standard deviation") + theme_classic() +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank()) +
  scale_x_continuous(limits=c(0, 700))

p3 <- temp %>%
  ggplot(aes(x=t, y=y_cv)) +
  geom_line(size=1.0) + geom_point()+
  ggtitle("coefficient of variation") + theme_classic()+
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank()) +
  scale_x_continuous(limits=c(0, 700))

p4 <- temp %>%
  ggplot(aes(x=t, y=y_skew)) +
  geom_line(size=1.0) + geom_point() +
  ggtitle("skewness") + theme_classic() +
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank()) +
  scale_x_continuous(limits=c(0, 700))

grid.arrange(p1, p2, p3, p4, ncol = 1)
```

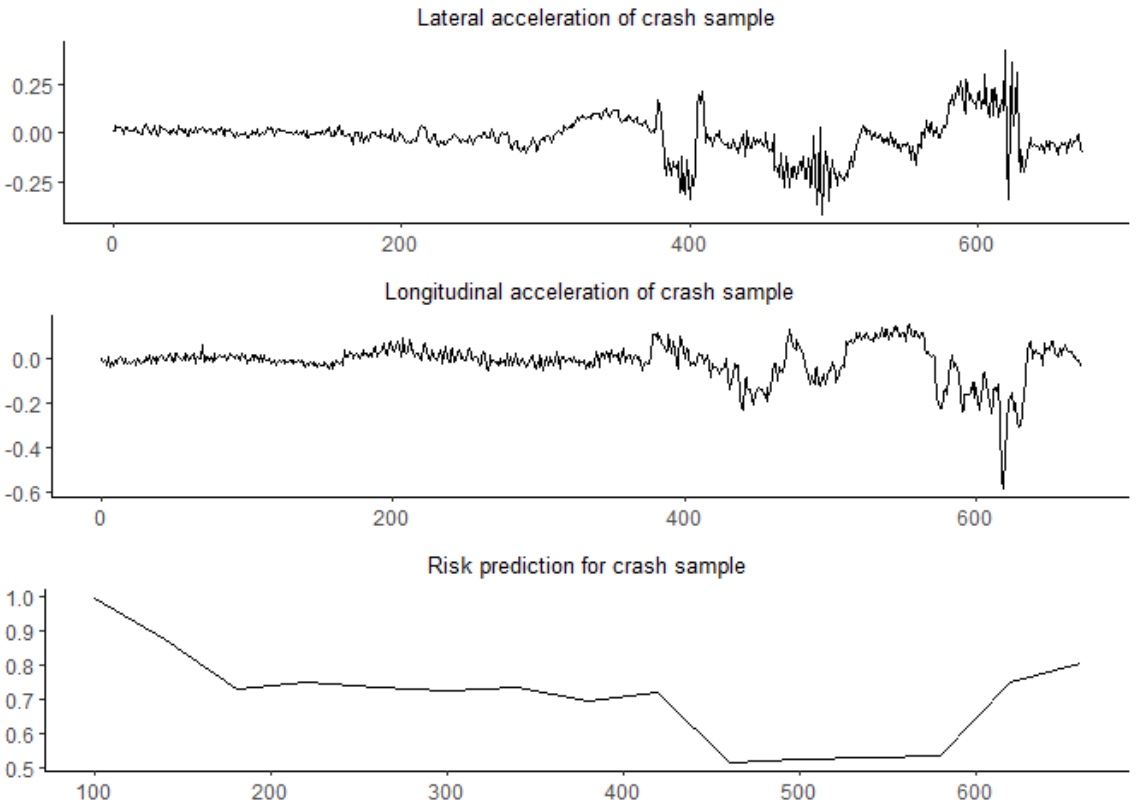


4.3 Online risk prediction

With the trained generalized addictive model and online surrogate estimates, we can predict the online risk for crash. We plot two trips with online risk in this section. These two trips come from crash and baseline separately. Both depict the online risk variation online the trip.

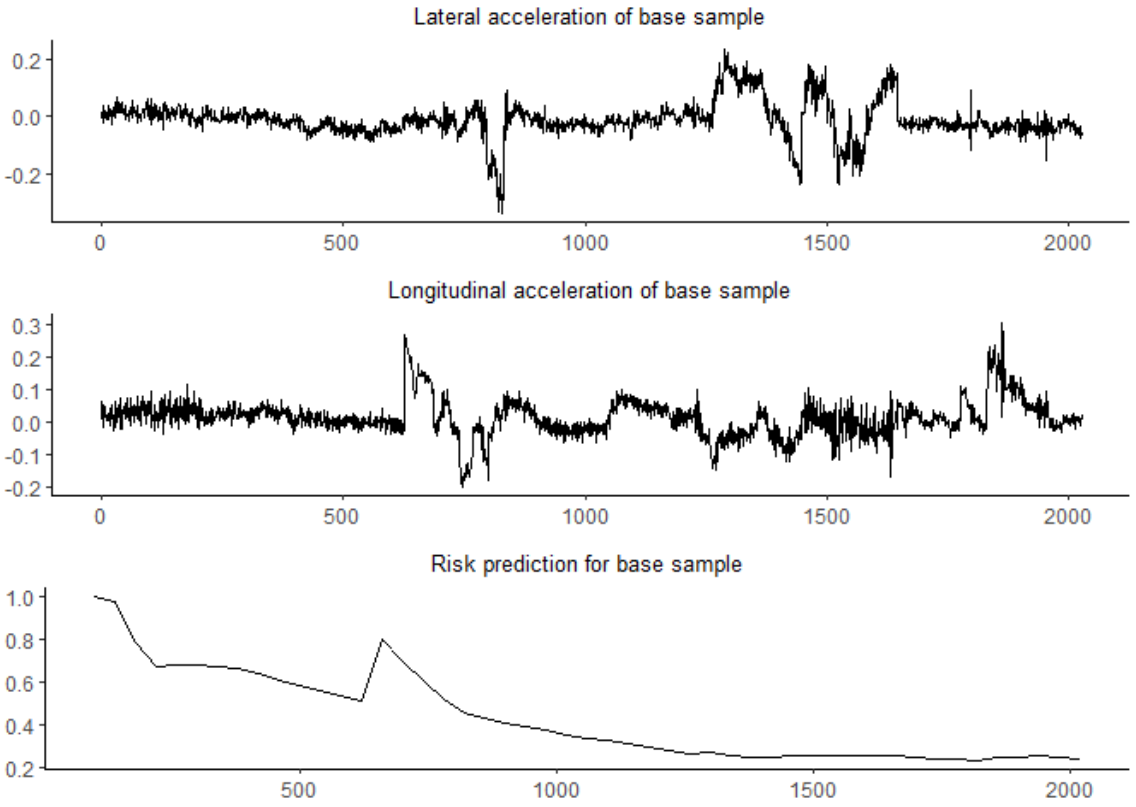
```
# crash sample risk prediction
dir_crash <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/trip/crash/"
setwd(dir_crash)
f_crash <- list.files(dir_crash)
trip_crash <- ldply(f_crash[17], read.csv, header=TRUE)


ac_std_x <- online_std(trip_crash$accel_x, 40, 100)
ac_std_y <- online_std(trip_crash$accel_y, 40, 100)
ac_cv_x <- online_cv(trip_crash$accel_x, 40, 100)
ac_cv_y <- online_cv(trip_crash$accel_y, 40, 100)
ac_ske_x <- online_skew(trip_crash$accel_x, 40, 100)
ac_ske_y <- online_skew(trip_crash$accel_y, 40, 100)
t <- c(0:(length(ac_std_x)-1))*40+100
temp <- data.frame(t, ac_std_x, ac_std_y,
                   ac_cv_x, ac_cv_y, ac_ske_x, ac_ske_y)

temp$risk <- predict(gam.sur, newdata = temp, type = "response")



p1 <- trip_crash %>%
  ggplot(aes(x=t, y=accel_y)) +
```

```
  geom_line(size=0.7) +
  ggtitle("Lateral acceleration of crash sample") + theme_classic()+
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())

p2 <- trip_crash %>%
  ggplot(aes(x=t, y=accel_x)) +
  geom_line(size=0.7) +
  ggtitle("Longitudinal acceleration of crash sample") + theme_classic()+
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())

p3 <- temp %>%
  ggplot(aes(x=t, y=risk)) +
  geom_line(size=0.7) +
  ggtitle("Risk prediction for crash sample") + theme_classic()+
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())

grid.arrange(p1, p2, p3, ncol = 1)
```



The above picture is online risk for crash samples. This online risk fluctuates as the change of lateral and longitudinal acceleration. In general, the risk is higher than 0.5, which indicates the high probability for crash.

```r
# base sample risk prediction
dir_base <- "C:/Users/yzhang/Desktop/appliedstat/OnlineSurrogate/data/trip/base/"
setwd(dir_base)
f_base <- list.files(dir_base)
trip_base <- ldply(f_base[17], read.csv, header=TRUE)


ac_std_x <- online_std(trip_base$accel_x, 40, 100)
ac_std_y <- online_std(trip_base$accel_y, 40, 100)
ac_cv_x <- online_cv(trip_base$accel_x, 40, 100)
ac_cv_y <- online_cv(trip_base$accel_y, 40, 100)
ac_ske_x <- online_skew(trip_base$accel_x, 40, 100)
ac_ske_y <- online_skew(trip_base$accel_y, 40, 100)
t <- c(0:(length(ac_std_x)-1))*40+100
temp <- data.frame(t, ac_std_x, ac_std_y,
                   ac_cv_x, ac_cv_y, ac_ske_x, ac_ske_y)

temp$risk <- predict(gam.sur, newdata = temp, type = "response")



p1 <- trip_base %>%
  ggplot(aes(x=t, y=accel_y)) +
  geom_line(size=0.7) +
  ggtitle("Lateral acceleration of base sample") + theme_classic()+
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())

p2 <- trip_base %>%
  ggplot(aes(x=t, y=accel_x)) +
  geom_line(size=0.7) +
  ggtitle("Longitudinal acceleration of base sample") + theme_classic()+
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())

p3 <- temp %>%
  ggplot(aes(x=t, y=risk)) +
  geom_line(size=0.7) +
  ggtitle("Risk prediction for base sample") + theme_classic()+
  theme(plot.title = element_text(size = 10, hjust = 0.5),
        axis.title.x=element_blank(),
        axis.title.y=element_blank())

grid.arrange(p1, p2, p3, ncol = 1)
```



The above picture is online risk for baseline samples. There is similar fluctuations for risk. In general, the risk becomes lower and lower as time goes by.

Conclusion

In this project, we proposed new surrogates for traffic safety analysis.

- Firstly, With a wide exploration of segments data for crash and baseline data. We find that the accelerations in three dimensions reveal the driving behaviors, which means it can serve as features for prediction of risk.
- Secondly, we propose three new surrogates-standard deviation, coefficient of variation and skewness. Compared with the surrogates proposed previously, our surrogates have better performance in four prediction models (Logistic regression, generalized addictive model, support vector machine, gradient boosting decision tree). Among these prediction models, GAM works best.
- Thirdly, we propose online estimation for surrogates. These online estimation varies as the accelerations change and provide online risk for crash as time goes by.

Reference

[1] Feng Guo. Statistical methods for naturalistic driving studies. Annual review of statisticsand its application, 6:309–328, 2019.

[2] Thomas A Dingus, Feng Guo, Suzie Lee, Jonathan F Antin, Miguel Perez, MindyBuchanan-King, and Jonathan Hankey. Driver crash risk factors and prevalence evalu-ation using naturalistic driving data. Proceedings of the National Academy of Sciences,113(10):2636–2641, 2016.

[3] Feng Guo, Sheila G Klauer, Jonathan M Hankey, and Thomas A Dingus. Near crashesas crash surrogate for naturalistic driving studies. Transportation Research Record,2147(1):66–74, 2010.

[4] Andrew P Tarko. Surrogate measures of safety. InSafe Mobility: Challenges, Method-ology and Solutions. Emerald Publishing Limited, 2018.