

```
## Load libraries
library(splines)
library(MASS)
library(xtable)

##Source functions
source("../functions.R")
```

Consider different numbers of features  $m$  and  $\lambda = 0.8$ :

```
m <- c(10, 100, 1000, 10000)
lambda <- 0.8

##save the maximum variance bound for each m and each scenario
maxSm <- matrix(NA, nrow=5, ncol=length(m))
```

## 1 Probability of being a false positive as a linear function of time

Get maximum of the variance bounds for different numbers of features:

```
for(i in 1:length(m))
{
  ntest <- m[i]

  tme <- seq(-1,2,length=ntest)

  maxSm[1,i] <- max(getVarBound(tme, lambda))
}
```

## 2 Probability of being a false positive as a smooth function of time

### 2.1 Linear fit

Get maximum of the variance bounds for different numbers of features:

```
for(i in 1:length(m))
{
  ntest <- m[i]
```

```
tme <- seq(-1,2,length=ntest)

maxSm[2,i] <- max(getVarBound(tme, lambda))
}
```

## 2.2 Spline fit

Get maximum of the variance bounds for different numbers of features:

```
for(i in 1:length(m))
{
  ntest <- m[i]

  tme <- seq(-1,2,length=ntest)
  splineMat <- ns(tme, df=3)

  maxSm[3,i] <- max(getVarBound(splineMat, lambda))
}
```

## 3 Probability of being a false positive as a sine + step function

### 3.1 Spline fit with df=3

Get maximum of the variance bounds for different numbers of features:

```
for(i in 1:length(m))
{
  ntest <- m[i]

  tme1 <- seq(-1*pi,2*pi,length=ntest)
  tme2 <- rep(1:0, each=ntest/2)

  splineMat3 <- cbind(ns(tme1,df=3), tme2)

  maxSm[4,i] <- max(getVarBound(splineMat3, lambda))
}
```

### 3.2 Spline fit with df=20

Get maximum of the variance bounds for different numbers of features. Note that an error is obtained for  $m = 10$  with the `getVarBound` function, which relies on `solve`, so use `ginv` instead:

```

for(i in 1:length(m))
{
  ntest <- m[i]

  tme1 <- seq(-1*pi,2*pi,length=ntest)
  tme2 <- rep(1:0, each=ntest/2)

  splineMat20 <- cbind(ns(tme1,df=20), tme2)

  x <- try(max(getVarBound(splineMat20, lambda)))
  if(class(x) == "numeric")
  {
    maxSm[5,i] <- x
  } else {
    print(paste("Error for i =", i, "with getVarBound, use ginv instead for matrix inverse"))

    zMat <- cbind(1, splineMat20)
    S <- zMat%*%ginv(t(zMat)%*%zMat)%*%t(zMat)
    maxSm[5,i] <- max(diag(S)/(4*(1-lambda)^2))
  }
}

## [1] "Error for i = 1 with getVarBound, use ginv instead for matrix inverse"

maxSm[5,]

## [1] 6.25000000 3.53984351 0.49085072 0.05085966

```

## 4 Table S1

```

colnames(maxSm) <- paste("m=",m,sep="")

xtable(maxSm, digits=3)

## % latex table generated in R 3.3.1 by xtable 1.8-2 package
## % Wed Jan 04 16:18:50 2017
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrrr}
## \hline
## & m=10 & m=100 & m=1000 & m=10000 \\
## \hline
## 1 & 2.159 & 0.246 & 0.025 & 0.002

```

```
## 2 & 2.159 & 0.246 & 0.025 & 0.002 \\
## 3 & 4.697 & 0.726 & 0.076 & 0.008 \\
## 4 & 4.798 & 0.755 & 0.079 & 0.008 \\
## 5 & 6.250 & 3.540 & 0.491 & 0.051 \\
## \hline
## \end{tabular}
## \end{table}
```

Session info:

```
devtools::session_info()

## Session info -----
## setting value
## version R version 3.3.1 (2016-06-21)
## system x86_64, mingw32
## ui RTerm
## language (EN)
## collate English_United States.1252
## tz America/New_York
## date 2017-01-04

## Packages -----
## package * version date source
## devtools 1.12.0 2016-06-24 CRAN (R 3.3.1)
## digest 0.6.9 2016-01-08 CRAN (R 3.3.1)
## evaluate 0.10 2016-10-11 CRAN (R 3.3.2)
## highr 0.6 2016-05-09 CRAN (R 3.3.1)
## knitr * 1.15.1 2016-11-22 CRAN (R 3.3.2)
## magrittr 1.5 2014-11-22 CRAN (R 3.3.1)
## MASS * 7.3-45 2016-04-21 CRAN (R 3.3.1)
## memoise 1.0.0 2016-01-29 CRAN (R 3.3.1)
## stringi 1.1.1 2016-05-27 CRAN (R 3.3.0)
## stringr 1.0.0 2015-04-30 CRAN (R 3.3.1)
## withr 1.0.2 2016-06-20 CRAN (R 3.3.1)
## xtable * 1.8-2 2016-02-05 CRAN (R 3.3.1)
```