Status Update 1

Samuel Spillane

March 23, 2020

Current Status and Plans

Currently, the status of my project is on track. I switched from using an SVM to a Naive Bayes classifier because there are more than two labels. I had thought that SVMs could work with more labels using kernelization or some other method, but this appears to not be the case. I am attempting a couple of things with my data, namely trying to classify using just the bone transformations, then just the bone positions, and finally the bone positions and transformations together in a data-frame. Currently, I have tested the transformations on their own with poor results. This is slightly expected given the invariance in the transformations. This invariance is mostly caused by the fact that Blender and most game engines use quaternions for rotations rather than eulerian angles. These being multiplicands of the complex quaternion function means that their domains are small. I am going to attempt different angular representations. I believe that including the bone positions into the features will not only improve accuracy but also make geospatial statistical modeling of the features easier.

Current Code

Here is my current work with the R-code:

```
#Load packages
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 3.5.3
```

```
library(e1071)
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.5.3
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
#Read data
poses<- read.csv("data.csv")

#For understanding the structure of the data.
str(poses)
```

```
## 'data.frame':    284 obs. of  12 variables:
##  $ Name  : Factor w/ 71 levels "breast_L","breast_R",..: 52 35 57 4 10 67 6 59 56 3 ...
##  $ PosX  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PosY  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ PosZ  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ RotW  : num  1 1 1 0.915 1 ...
##  $ RotX  : num  0 0 0 -0.277 0 ...
##  $ RotY  : num  0 0 0 0.28 0 ...
##  $ RotZ  : num  0 0 0 0.086 0 ...
##  $ ScaleX: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ ScaleY: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ ScaleZ: num  1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ Pose  : Factor w/ 4 levels "run","sitting",..: 4 4 4 4 4 4 4 4 4 4 ...
```

```
#For understanding the distributions of the data
describe(poses)
```

```
##        vars   n  mean    sd median trimmed   mad   min   max range  skew
## Name*     1 284 36.00 20.53   36.0    36.0 26.69  1.00 71.00 70.00  0.00
## PosX      2 284  0.00  0.00    0.0     0.0  0.00  0.00  0.00  0.00   NaN
## PosY      3 284  0.00  0.00    0.0     0.0  0.00  0.00  0.00  0.00   NaN
## PosZ      4 284  0.00  0.00    0.0     0.0  0.00  0.00  0.00  0.00   NaN
## RotW      5 284  0.99  0.04    1.0     1.0  0.00  0.69  1.00  0.31 -4.86
## RotX      6 284  0.01  0.11    0.0     0.0  0.00 -0.68  0.63  1.31  2.28
## RotY      7 284  0.00  0.04    0.0     0.0  0.00 -0.39  0.34  0.73  1.48
## RotZ      8 284  0.00  0.05    0.0     0.0  0.00 -0.43  0.35  0.77 -1.04
## ScaleX    9 284  1.00  0.00    1.0     1.0  0.00  1.00  1.00  0.00   NaN
## ScaleY   10 284  1.00  0.00    1.0     1.0  0.00  1.00  1.00  0.00   NaN
## ScaleZ   11 284  1.00  0.00    1.0     1.0  0.00  1.00  1.00  0.00   NaN
## Pose*    12 284  2.50  1.12    2.5     2.5  1.48  1.00  4.00  3.00  0.00
##        kurtosis   se
## Name*     -1.21 1.22
## PosX        NaN 0.00
## PosY        NaN 0.00
## PosZ        NaN 0.00
## RotW      24.22 0.00
## RotX      23.69 0.01
## RotY      47.13 0.00
## RotZ      37.83 0.00
## ScaleX      NaN 0.00
## ScaleY      NaN 0.00
## ScaleZ      NaN 0.00
## Pose*     -1.37 0.07
```

```
#Set seed so that data can be easily reproduced
set.seed(762)

#Split data and format it for model, may subset data based on bone name later depending on results
splitIndex = createDataPartition(y = poses$Pose,p = 0.75,list = FALSE)
train = poses[splitIndex,]
test = poses[-splitIndex,]

#Train naive-bayes
model = naiveBayes(Pose~., data=train)
print(model)
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     run sitting   tpose    walk
##    0.25    0.25    0.25    0.25
##
## Conditional probabilities:
```

```
##          Name
## Y           breast_L   breast_R     calf_L     calf_R calf_twist_L
##   run     0.01851852 0.01851852 0.01851852 0.00000000   0.01851852
##   sitting 0.01851852 0.00000000 0.01851852 0.01851852   0.00000000
##   tpose   0.00000000 0.01851852 0.01851852 0.00000000   0.01851852
##   walk    0.00000000 0.00000000 0.00000000 0.00000000   0.01851852
##          Name
## Y         calf_twist_R clavicle_L clavicle_R     foot_L     foot_R
##   run       0.00000000 0.01851852 0.01851852 0.01851852 0.01851852
##   sitting   0.01851852 0.01851852 0.00000000 0.01851852 0.01851852
##   tpose     0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##   walk      0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##          Name
## Y             hand_L     hand_R       head  index00_L  index00_R
##   run     0.01851852 0.01851852 0.00000000 0.00000000 0.01851852
##   sitting 0.01851852 0.01851852 0.01851852 0.01851852 0.00000000
##   tpose   0.00000000 0.01851852 0.01851852 0.00000000 0.01851852
##   walk    0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##          Name
## Y          index01_L  index01_R  index02_L  index02_R  index03_L
##   run     0.01851852 0.00000000 0.00000000 0.01851852 0.01851852
##   sitting 0.01851852 0.01851852 0.01851852 0.01851852 0.00000000
##   tpose   0.00000000 0.01851852 0.01851852 0.01851852 0.01851852
##   walk    0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##          Name
## Y          index03_R lowerarm_L lowerarm_R lowerarm_twist_L
##   run     0.01851852 0.01851852 0.01851852       0.01851852
##   sitting 0.00000000 0.01851852 0.00000000       0.01851852
##   tpose   0.01851852 0.00000000 0.01851852       0.01851852
##   walk    0.01851852 0.01851852 0.01851852       0.00000000
##          Name
## Y         lowerarm_twist_R middle00_L middle00_R middle01_L middle01_R
##   run           0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##   sitting       0.00000000 0.01851852 0.01851852 0.01851852 0.01851852
##   tpose         0.01851852 0.01851852 0.00000000 0.01851852 0.01851852
##   walk          0.01851852 0.00000000 0.00000000 0.01851852 0.01851852
##          Name
## Y         middle02_L middle02_R middle03_L middle03_R       neck
##   run     0.00000000 0.01851852 0.01851852 0.01851852 0.01851852
##   sitting 0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##   tpose   0.00000000 0.00000000 0.01851852 0.00000000 0.01851852
##   walk    0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##          Name
## Y             pelvis  pinky00_L  pinky00_R  pinky01_L  pinky01_R
##   run     0.00000000 0.01851852 0.01851852 0.00000000 0.01851852
##   sitting 0.01851852 0.00000000 0.01851852 0.01851852 0.00000000
##   tpose   0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##   walk    0.01851852 0.00000000 0.01851852 0.01851852 0.01851852
##          Name
## Y          pinky02_L  pinky02_R  pinky03_L  pinky03_R   ring00_L
##   run     0.01851852 0.01851852 0.01851852 0.00000000 0.01851852
##   sitting 0.01851852 0.01851852 0.01851852 0.01851852 0.00000000
##   tpose   0.01851852 0.01851852 0.01851852 0.01851852 0.00000000
##   walk    0.00000000 0.00000000 0.01851852 0.01851852 0.00000000
```

```
##          Name
## Y           ring00_R   ring01_L   ring01_R   ring02_L   ring02_R
##   run     0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##   sitting 0.01851852 0.00000000 0.00000000 0.01851852 0.01851852
##   tpose   0.01851852 0.01851852 0.00000000 0.01851852 0.01851852
##   walk    0.01851852 0.01851852 0.01851852 0.01851852 0.00000000
##          Name
## Y           ring03_L   ring03_R       root    spine01    spine02
##   run     0.00000000 0.00000000 0.01851852 0.01851852 0.01851852
##   sitting 0.01851852 0.00000000 0.01851852 0.01851852 0.00000000
##   tpose   0.00000000 0.01851852 0.00000000 0.00000000 0.01851852
##   walk    0.00000000 0.01851852 0.00000000 0.01851852 0.01851852
##          Name
## Y            spine03    thigh_L    thigh_R thigh_twist_L thigh_twist_R
##   run     0.01851852 0.01851852 0.00000000    0.00000000    0.01851852
##   sitting 0.01851852 0.01851852 0.01851852    0.01851852    0.00000000
##   tpose   0.01851852 0.01851852 0.01851852    0.01851852    0.01851852
##   walk    0.01851852 0.01851852 0.00000000    0.01851852    0.01851852
##          Name
## Y          thumb01_L  thumb01_R  thumb02_L  thumb02_R  thumb03_L
##   run     0.00000000 0.01851852 0.01851852 0.01851852 0.01851852
##   sitting 0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##   tpose   0.01851852 0.01851852 0.01851852 0.01851852 0.01851852
##   walk    0.00000000 0.01851852 0.01851852 0.01851852 0.01851852
##          Name
## Y          thumb03_R     toes_L     toes_R upperarm_L upperarm_R
##   run     0.01851852 0.01851852 0.00000000 0.01851852 0.01851852
##   sitting 0.01851852 0.01851852 0.01851852 0.01851852 0.00000000
##   tpose   0.01851852 0.00000000 0.01851852 0.00000000 0.01851852
##   walk    0.01851852 0.00000000 0.01851852 0.01851852 0.01851852
##          Name
## Y          upperarm_twist_L upperarm_twist_R
##   run           0.00000000       0.01851852
##   sitting       0.01851852       0.01851852
##   tpose         0.01851852       0.01851852
##   walk          0.01851852       0.01851852
##
##          PosX
## Y          [,1] [,2]
##   run         0    0
##   sitting     0    0
##   tpose       0    0
##   walk        0    0
##
##          PosY
## Y          [,1] [,2]
##   run         0    0
##   sitting     0    0
##   tpose       0    0
##   walk        0    0
##
##          PosZ
## Y          [,1] [,2]
##   run         0    0
```

```
##   sitting    0    0
##   tpose      0    0
##   walk       0    0
##
##          RotW
## Y             [,1]       [,2]
##   run     0.9923575 0.03407907
##   sitting 0.9746540 0.06492339
##   tpose   1.0000000 0.00000000
##   walk    0.9971736 0.01339408
##
##          RotX
## Y               [,1]       [,2]
##   run     0.010962580 0.08431183
##   sitting 0.045406345 0.19014838
##   tpose   0.000000000 0.00000000
##   walk    0.003489461 0.03495236
##
##          RotY
## Y                [,1]       [,2]
##   run      0.013804561 0.06172808
##   sitting -0.006074477 0.05500089
##   tpose    0.000000000 0.00000000
##   walk     0.004022623 0.02267605
##
##          RotZ
## Y                 [,1]       [,2]
##   run      0.0002342548 0.05583045
##   sitting -0.0158062627 0.07214215
##   tpose    0.0000000000 0.00000000
##   walk     0.0006210089 0.06169967
##
##          ScaleX
## Y         [,1] [,2]
##   run        1    0
##   sitting    1    0
##   tpose      1    0
##   walk       1    0
##
##          ScaleY
## Y         [,1] [,2]
##   run        1    0
##   sitting    1    0
##   tpose      1    0
##   walk       1    0
##
##          ScaleZ
## Y         [,1] [,2]
##   run        1    0
##   sitting    1    0
##   tpose      1    0
##   walk       1    0
```

```
trainPred=predict(model, newdata = train, type = "class")
trainTable=table(train$Pose, trainPred)
testPred=predict(model, newdata=test, type="class")
testTable=table(test$Pose, testPred)

#Get accuracy
trainAcc=(trainTable[1,1]+trainTable[2,2]+trainTable[3,3])/sum(trainTable)
testAcc=(testTable[1,1]+testTable[2,2]+testTable[3,3])/sum(testTable)
#Contingency Table
print(trainTable)
```

```
##          trainPred
##           run sitting tpose walk
##   run       1       2    51    0
##   sitting   0       7    47    0
##   tpose     0       0    54    0
##   walk      1       2    50    1
```

```
print(testTable)
```

```
##          testPred
##           run sitting tpose walk
##   run       0       1    16    0
##   sitting   0       2    15    0
##   tpose     0       0    17    0
##   walk      1       1    15    0
```

```
#Print Accuracy
print(round(cbind(trainAccuracy=trainAcc, testAccuracy=testAcc),3))
```

```
##      trainAccuracy testAccuracy
## [1,]         0.287        0.279
```

Reflection

As can be seen, the accuracy is currently less than 30%. This is terrible. My plan for the next week is to find ways to improve this. I have outlined some of the changes I am making above, but I am open to further suggestions.