Hand Gesture Data Analysis

Lincoln Wells

December 3, 2014

Stat 407, Final Project

Description of the Data

The data that will be analyzed within this report is related to sign language and how different aspects of the signer's hands change as a person signs a word. The data was collected by researchers who are interested in whether or not readings from an inexpensive depth camera and open source computer libraries can be used to collect data from a signer and then predict what word the signer is saying. The variables that the cameras were able to capture include the x, y, and z coordinates of the hand's centroid, the hand's bounding ellipse two dimensional height, width and angle, and the x-y area of each hand as it signs. Therefore, the data set contains 7 explanatory variables.

The camera was able to collect this information through a controlled experiment. The experiment begins when the signer initiated a gesture recognition window. The camera then begins taking snapshots of the signer periodically throughout the entirety of the sign. After the sign is completed, the library system uses a skin detection algorithm, which contains noise-reducing morphologies, to identify the user's hands, face, and other features separately. The software libraries then take the measurements of the hands only and creates the variables that are described above.

This procedure was repeated and data was collected on four different individuals. Each individual was asked to sign two different words left and right and the subjects signed each word 5 times. Therefore snapshots were taken on a total of 40 signs. The camera took a total of 1015 snapshots over these 40 signs, but it did not take the same amount of snapshots for each trial.

Primary Questions

Main Question: Using only an inexpensive depth camera and open source computer vision libraries, can a computer recognize a word that a person does in American Sign Language?

Subsequent Questions: If this recognition process is possible, will the accuracy of the recognition vary significantly between participants?

Can a model be created that will accurately recognize which sign is being performed for any participant?

Plan for the Analysis

The first step in the data analysis is creating a data set that is usable in future analysis. The problem with the current data set is that each individual repetition contains a different number of observations, or snapshots. In other words, the camera took more pictures for some repetitions than it did for others. An example of this can be found below:

```
asl<-read.csv("ASL2new.csv")
ASL61 <- asl[ which(asl$Gesture=='left' & asl$User=='ASL6'& asl$Rep=='1'), ]
dim(ASL61)
```

```
## [1] 26 12
```

```
ASL62 <- asl[ which(asl$Gesture=='left' & asl$User=='ASL6'& asl$Rep=='2'), ]
dim(ASL62)
```

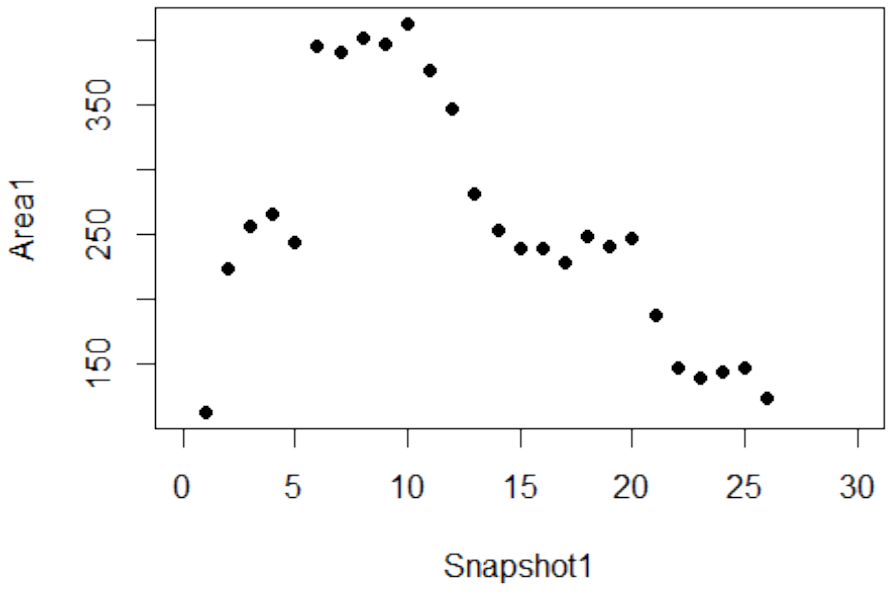
```
## [1]  9 12
```

This example demonstrates that the participant labeled ASL6 is doing the gesture "left". During the first repition of this, the camera was able to 26 pictures of the signing process, but during the second repition the camera only took 9 pictures. This is a major problem in the data because the observations cannot be compared to one another.

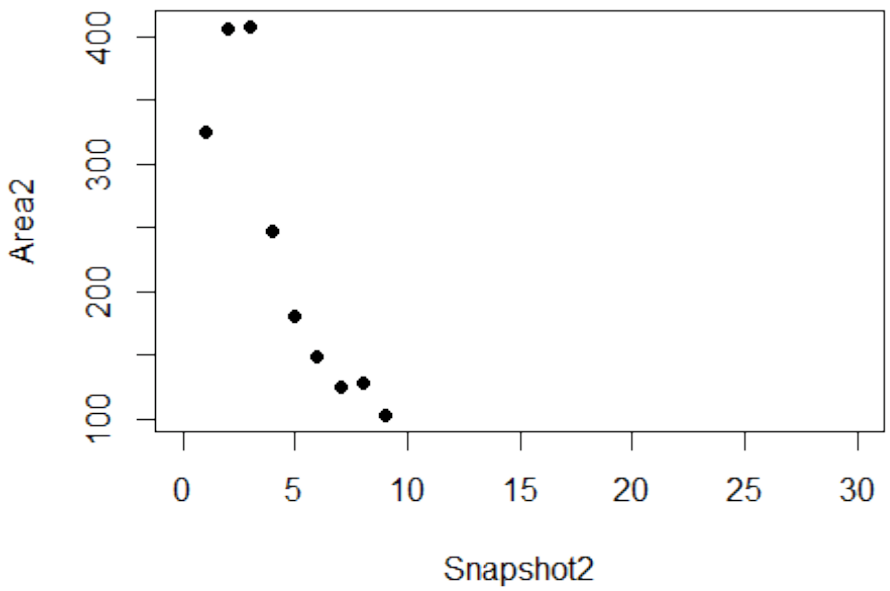
For example, the 9th observation cannot be compared in the example listed above. For the first repition, the 9th observation, or snapshot taken, represents the hand position of the sign when it is 34.6% complete (9/26(100%)). For the second repition, the 9th observation represents the hand position of the sign when it is 100% complete (9/9(100%).

This can also be represented graphically:

```
Snapshot1<-ASL61$Snapshot
Area1<-ASL61$Area
plot(Snapshot1, Area1, pch=19, xlim=c(0,30))
```




```
Snapshot2<-ASL62$Snapshot
Area2<-ASL62$Area
plot(Snapshot2, Area2, pch=19, xlim=c(0,30))
```



Because these two observations have differing x ranges, the first step required in the data analysis is standardizing the data on the same x axis so they can be compared to one another in future analysis.

After the observations are set on the same x range, a smoothed line needs to be fitted to each individual repetition. This smoothed line will be used to predict the variable value, which in the case above was area, for set values which represent how far through the signing process each repition is. This could be done by predicing the values of 10%, 20%, 30%, ... , and 90% of the way through the process for each repition. This would give us 9 variables of area for each repition. These nine variables would be able to be compared throughout the repitions because they all represent what the area of the hand is 10% of the way through the sign, which should be the same if the gesture is the same.

This process would create a data set of 40 observations, one for each repition, and 63 explanatory varaibles, 9 for each of the original 7. At this point, a data set would be created that can be used for further analysis to help recognize which gesture the person is signing.

Now that a new data set that has comparable observations is created, the analysis of using the data to predict and recognize the gestures can begin. To begin, this analysis, the data will need to be plotted on to see if the data comes from a mutivariate normal distribution. If the data is skewed, it means that it may not have come from a normal distribution. After this is performed, a calculation of variance-covariance will need to be performed. The variance-covariances need to be determined the same or not before the analysis can continue.

If the variance-covariances are equal and the data comes from a normal distribution, continue the analysis by preforming PLDA analysis to predict which gesture is being preformed.If either of these condidionts are not meet, PLDA cannot be preformed on this data set. Regardless of the outcome outlined above, continue the analysis by preforming only Classification Trees analysis to predict which gesture is being preformed.

Based on the outcome of the previous steps, PLDA or Classification Trees can then be used to predict which gesture is being made based on the available data captured by the depth camera. After this is completed, the error of the model can be calculated to answer whether or not the analysis is doing a good job of recognizing which gesture is being performed.

Lastly, the data will be summarized through summary statistics, plots, descriptions of the model, and calculations of error that will demonstrate whether or not the model is recognizing the gesture. The summary will focus on visual interpretations of the finding to easily deomonstrate how well the model is working.

Initial Analysis, Creation of the Data Set and Checking Conditions

To begin the analysis, a new data set needs to be created because the variables need to be set on the same x range (this is described above). To do this, a loess analysis was preformed to fit a smooth line through the data points for each repition. Then, 9 points were predicted at each .1 interval of the loess smooth curve. These new predicted values were then used to create 40 observations which contains 66 variables, the three original categorical variables, and the 63 new explanatory variables that were created (7 explanatory variables* 9 time intervals).

However, there was a significant issue with the original loess analysis. One of the participants, ASL9, appears to be a very fast signer. Therefore, the camera was unable to take enough snapshots for the loess analysis to be preformed (at least 10 observations are necessary to perfrom the loess analysis, but ASL9 was able to complete the sign with as few as 2 observations recorded). Overall, 4 of ASL9's 10 repitions were unable to have the loess analyisis preformed on it, and therefore ASL9's observations were dropped from the experiment. Another observation (ASL6's second repitions) also needed to be dropped because it contained less than 10 observations, but because it still contained 9 other observations, ASL6 was allowed to remain in the experimental data.

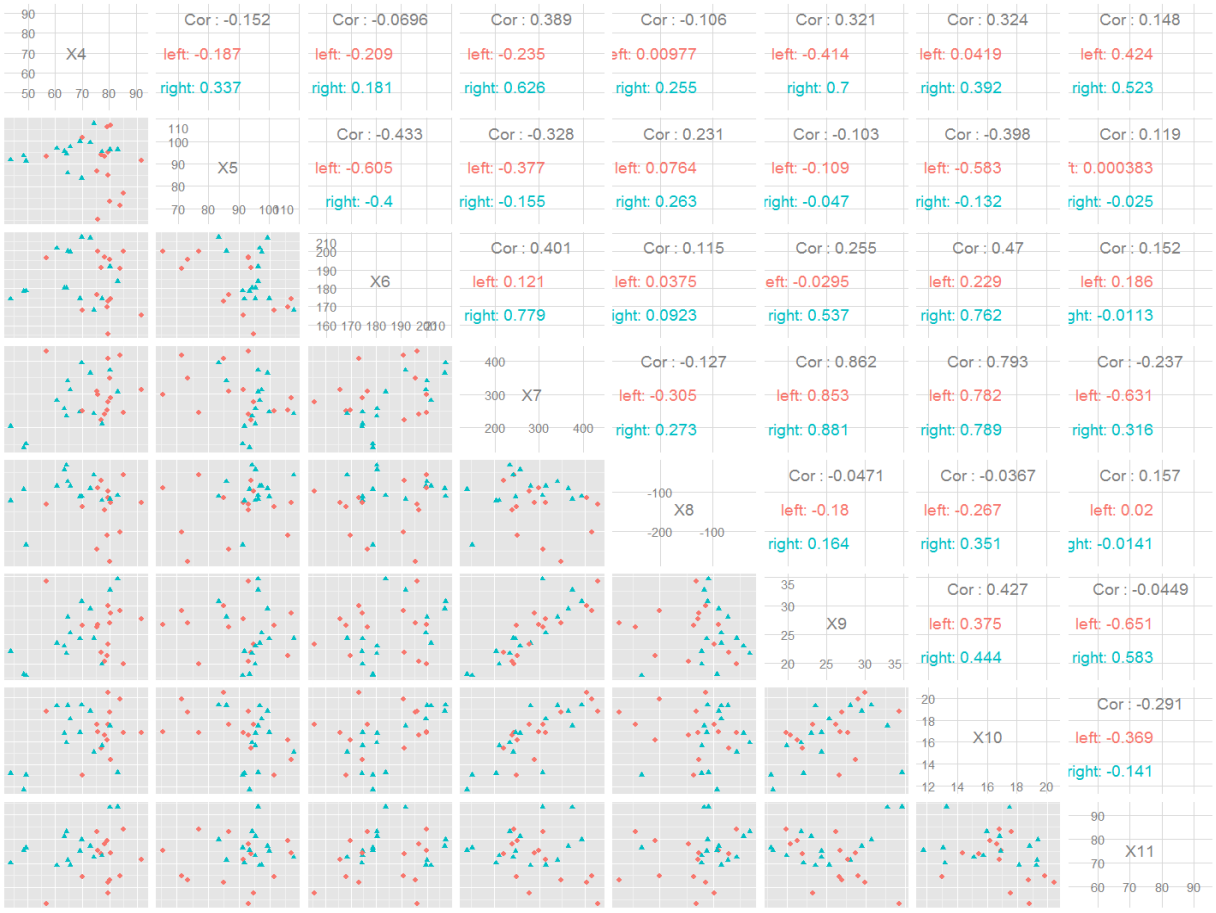
The process for the creation of the new data set using the loess analysis can be found before. The data set formed from this analysis is called ASL.s.wide. This data set was created using ths csv file "ASL2woUSERASL9" which is the original data set but omitting the user ASL9.

This code can be found below in the appendix, page 1.

Now a data set has been created where each repitition is an observation (there were originally 40 reptitions, then we subtracted all 10 from user ASL9 and 1 from ASL 6 to obtain 29 observations), and each of the 7 explanatory variables are now split into 9 time intervals creating 63 explanatory variables plus the 3 categorical variables.

The next step is to check if the conditions of normality is met to see if we can preform PLDA analysis. Because there are 63 explanatory, a scatterplot matrix which contains every variable cannot realistically be created. Therefore, the first seven variables were compared to see if there was an obvious trend of anormality within these first set of variables.

```
library(GGally)
ggpairs(ASL.s.wide, columns=4:11, colour="X1", shape="X1")
```



The scatterplots and its shapes suggest that this data from each of the sign language gestures types can't be considered to be a sample that comes from a multivariate normal. This is true because the scatter plots are not elliptical and also because the spread of the points is different. This is very obvious in the comparision of X4 and X5 where right spread horizontally and left is spread vertically within this comparision, however this type of spread can be found on many of the scatterplots.

Because the data does not come from a multivariate normal, PLDA cannot be preformed to predict which gesture is being preformed without significant transformations to the data, which will not be apart of this study.

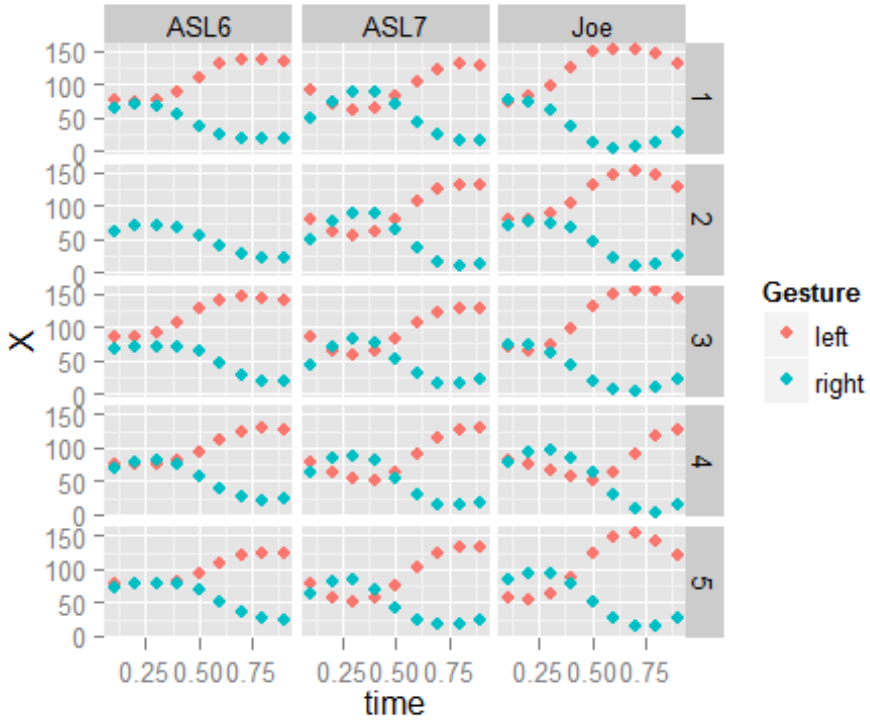
Prediction and Classification of the Observations

Now that we have determined we are not able to preform the PLDA analysis, we can prefrom the Classificant Trees Analysis and calculate the error that this model gives to determine if it can accuratly predict which gesture is being preformed and which repitions it is not classifying correctly.

This code can be found below in the appendix, page 2.

The error for the Classification Trees Analysis is 0/29=0%. Therefore there were no values in the test or training set that were misclassified. This was confirmed using R's predict function. The tree function works by using X39, which corresponds to the X centroid of the gesture during the 6th time interval. This can be viewn graphically by looking at the .6 value on the follwing scatterplot matrix:

```
qplot(time, X, data=ASL.s, colour=Gesture) + facet_grid(Rep~User)
```



This graph clearly distinguishes between the left and the right gesture. This is why it was used in the classification tree to distinguish between the two types of gestures because the difference between the two gestures for this variable is constant between all of the participants.

Conclusions

Because the Classification Tree Analysis produced a model that has 0% error in the test data, it is reasonable to say that an inexpensive depth camera and open source computer

vision libraries can be used to recognize a word that a person does in American Sign Language. This can be done by looking at the X-centroid value of the gesture when it is 60% completed. If this value is above 57.15, then it is considered to be the word left and otherwise it is classified as the word right.

It is also reasonable to assume that the accuracy of the recognition does not vary significantly between the participants of the study. This is because the model produced 0% error meaning that it was able to recognize the gesture for every individual participant. It may also be reasonable to say that this model can be used on any individual for the same reason as before with the 0% error found on any individual.

In conclusion, the classification tree created a model that can accurately recognize whether a person is doing the sign language gesture left or right with 0% error. This model created no variation between individual participants and it may be able to be used on other participants in the future because of the constant difference in the X-centroid of each gesture.

Appendix: Page 1

```
library(ggplot2)
ASL2 <- read.csv("ASL2woUSERASL9.csv")
# Get the same number of points for each rep, user, gesture
ASL.s <- data.frame(Gesture=NULL, User=NULL, Rep=NULL, X=NULL, Y=NULL,
Z=NULL, Area=NULL, Angle =NULL, Height=NULL, Width=NULL, time=NULL)
for (i in unique(ASL2$User))
  for (j in unique(ASL2$Rep))
    for (k in unique(ASL2$Gesture)) {
      x <- subset(ASL2, User==i&Rep==j&Gesture==k)
      x$time <- (1:nrow(x))/nrow(x)
      cat(nrow(x), i, j, k, "\n")
      if (nrow(x)>9) {
      x.l <- data.frame(x[1:9,1:10], time=seq(0.1, 0.9, 0.1))
      for (ii in 4:10) {
        f <- as.formula(paste(colnames(x)[ii],"~time"))
        l <- loess(f, data=x)
        x.l[,ii] <- predict(l, x.l$time)
      }
      ASL.s <- rbind(ASL.s, x.l)
    }}

# Rearrange data, to have one column for each time point
ASL.s.wide <- data.frame(matrix(0, 29, 63+3))
ASL.s.wide[,1:3] <- unique(ASL.s[,1:3])
ASL.s.wide[,4:10] <- ASL.s[ASL.s$time==0.1, 4:10]
ASL.s.wide[,11:17] <- ASL.s[ASL.s$time==0.2, 4:10]
ASL.s.wide[,18:24] <- ASL.s[abs(ASL.s$time-0.3)<0.01, 4:10]
ASL.s.wide[,25:31] <- ASL.s[ASL.s$time==0.4, 4:10]
ASL.s.wide[,32:38] <- ASL.s[ASL.s$time==0.5, 4:10]
ASL.s.wide[,39:45] <- ASL.s[ASL.s$time==0.6, 4:10]
ASL.s.wide[,46:52] <- ASL.s[abs(ASL.s$time-0.7)<0.01, 4:10]
ASL.s.wide[,53:59] <- ASL.s[ASL.s$time==0.8, 4:10]
ASL.s.wide[,60:66] <- ASL.s[ASL.s$time==0.9, 4:10]
dim(ASL.s.wide)

## [1] 29 66
```

Appendix: Page 2

```
ASL.s.wide$X2<-NULL
set.seed(50)
ASL2.tr <- ASL.s.wide[
which(ASL.s.wide$X3=='3'|ASL.s.wide$X3=='4'|ASL.s.wide$X3=='5'), ]
ASL2.ts <- ASL.s.wide[ which(ASL.s.wide$X3=='1'|ASL.s.wide$X3=='2'), ]
ASL2.tr$X3<-NULL
ASL2.ts$X3<-NULL

library(rpart)
ASL2.rp<-rpart(X1~., data=ASL2.tr[,-2], control=rpart.control(minsplit=2))
ASL2.rp
```

```
## n= 18
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 18 9 left (0.5000 0.5000)
##   2) X39>=57.15 9 0 left (1.0000 0.0000) *
##   3) X39< 57.15 9 0 right (0.0000 1.0000) *
```

```
table(ASL.s.wide$X1,predict(ASL2.rp, ASL.s.wide,type = "class"))
```

```
##
##         left right
##   left    14     0
##   right    0    15
```

```
#predict(ASL2.rp, ASL.s.wide,X1="class")
```