Detecting Breast Tumors using Mammograms

William F Lamberti

STAT 763

Dr. Carr

Abstract

Mammograms are the main first method for detecting if patients have breast cancer. Computer aided detection (CAD) can be helpful in aiding a radiologist in diagnosis in identifying concerning features in a mammogram. However, CAD has some controversies in regards to producing meaningful results. Even if CAD does help, the rates of correctly identifying patients without and without malignant tumors is somewhat low. Thus, improvements in correctly distinguishing between those with and without tumors is warranted. For the analysis of this paper, the gray scale intensities of mammogram images and various transformation where used in an attempt to distinguish between those with tumors and those without tumors. Methods utilized included, but were not limited to, SVM, PCA and K-means.

Introduction

Mammograms are x ray images of an individual's breasts and are used to detect cancer. The general process is that each breast is placed between two surfaces that squeezes the breasts. This is done so that the tissues are spread out as much as possible. Then, an image is taken and analyzed for the presence of cancer. This is repeated for each breast on its own. Image 1 shows an artist's rendition of a women having a mammogram[1]. There are many benefits of mammograms. They are cheap and noninvasive data collection method that works well, only expose the patient to a low amount of radiation, are completely covered by most health care plans, and are a fairly standard practice. However, using mammograms does not guarantee detection for a variety of reasons. Both methods of analysis, using computer aided detection (CAD) or not, both pose different challenges which we will address shortly. The type of breasts a patient has does effect analysis. For example, having fatty breasts eases the analysis and will have higher values for sensitivity and specificity. However, having denser breasts complicates diagnosis. This results

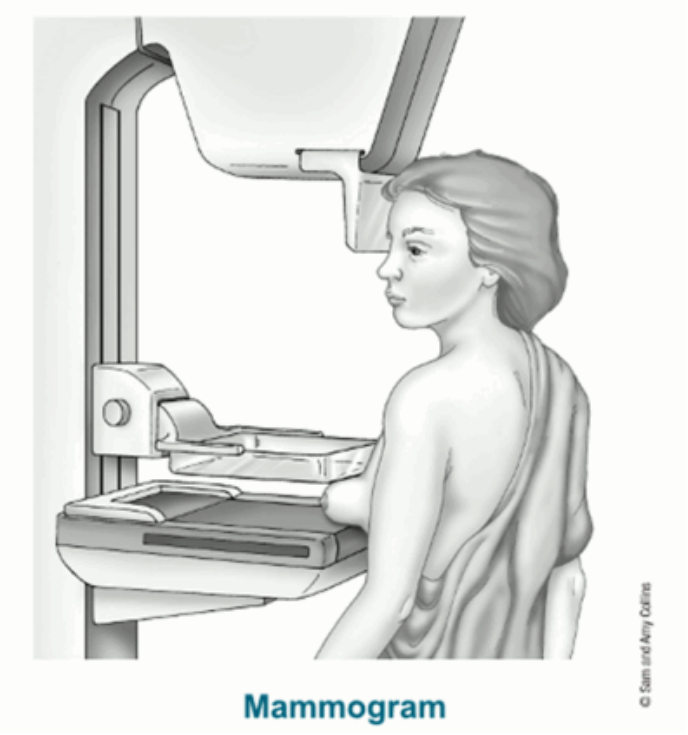

Figure 1

[1] "How Is a Mammogram Done?" How Is a Mammogram Done? Accessed April 18, 2016. http://www.cancer.org/healthy/findcancerearly/examandtestdescriptions/mammogramsandotherbreastimagingprocedures/mammograms-and-other-breast-imaging-procedures-having-a-mammogram.

Figure 2

in lower values for sensitivity and specificity. See Image 2 for the four levels of breast density[2]. In short, 2D mammograms do pose some shortcomings.

There are two primary methods of analyzing the mammograms. The first is that a radiologist will visually try to detect any concerning features without the use of a computer. In general, a radiologist is looking for abnormalities. This includes features such as highly dense white areas. They are looking for large concentrations of white because features, such as tumors, that are not fatty will stop more of the x rays and appear whiter. One common method utilized is comparing the breasts side by side. If the breasts appear symmetric, then this is a positive sign that both breasts are normal. If one breast appears to have large white areas that another does not have, then this warrants further analysis. While I have not encountered a name for this method, I will need to reference this specific method again, so I will call it the Comparison Method.

[2] "Mammogram." Breast Density — The Four Levels. Accessed April 18, 2016. http://www.mayoclinic.org/tests-procedures/mammogram/multimedia/breast-density-mdash-the-four-levels/img-20008862.

However, there are possible exceptions to this. For example, if a patient had a procedure on their breasts, any scarred tissues would also be bright white. Thus, the shape, density, central masses, and other features are helpful in properly diagnosing abnormalities.[3] Sometimes two radiologists will analyze the images, but this is not always possible.

The alternative method uses CAD. CAD is used for decreasing false negative rates. The recommended procedure is to have a single radiologist perform a review as normal. Then, a second review will be performed by CAD. It will highlight and bring concerning abnormalities to the attention of the radiologist. Then, the radiologist must decide whether or not the abnormality warrants further investigation.[4] CAD is more capable of detecting ductal carnimoa in situ, or 'stage 0' cancer. CAD has been found to identify more potential cases of cancer, and therefore save more live. According to one of the first studies performed to see how well CAD performed in comparison to a professional radiologist, CAD use helped to correctly identify malignant tumors from 73.5% to 87.4%. It also increase the number of correctly identified those who do not have malignant tumors from 31.6% to 41.9%.[5] Obviously, the low rate of correctly classifying those with malignant tumors is still not at an desirable. This in turn, has the patient undergo several more tests and procedures. Starting with the obvious, it has more patients undergo additional tests. Some of these patients must have invasive tests like a biopsy. It also puts additional economic and physiological strain on the patient. It is not guaranteed that these tests are covered by health insurance or are affordable with insurance. Thus, when using CAD

---

[3] "Mammogram Examples and Ultrasound Breast Pictures." Breast Cancer. 2016. Accessed April 18, 2016. http://breast-cancer.ca/mammims/.
[4] Castellino, Ronald A. "Computer Aided Detection (CAD): An Overview." Cancer Imaging. Accessed April 18, 2016. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1665219/.
[5] Wu, Qiu, and Mia Markey. "Computer-Aided Diagnosis of Breast Cancer on MR Imaging." *Recent Advances in Breast Imaging, Mammography, and Computer-Aided Diagnosis of Breast Cancer*: 739-62. doi:10.1117/3.651880.ch22.

results in a false positive, it can cause unneeded strain on the patient. According to a study performed in 2013, CAD use has increased from about 4% in 2001 to about 61% in 2006.[6] This study also stated that CAD use results in additional tests, breast biopsies, DCIS diagnoses, early stage invasive cancer diagnoses, and false positives.[7] Other experts guess that CAD is making radiologists too reliant on technology and not as prudent. Thus, while the use of CAD is somewhat complicated, improving upon the optimal solution, having two radiologists analyze the mammograms, is still warranted.[8]

Data

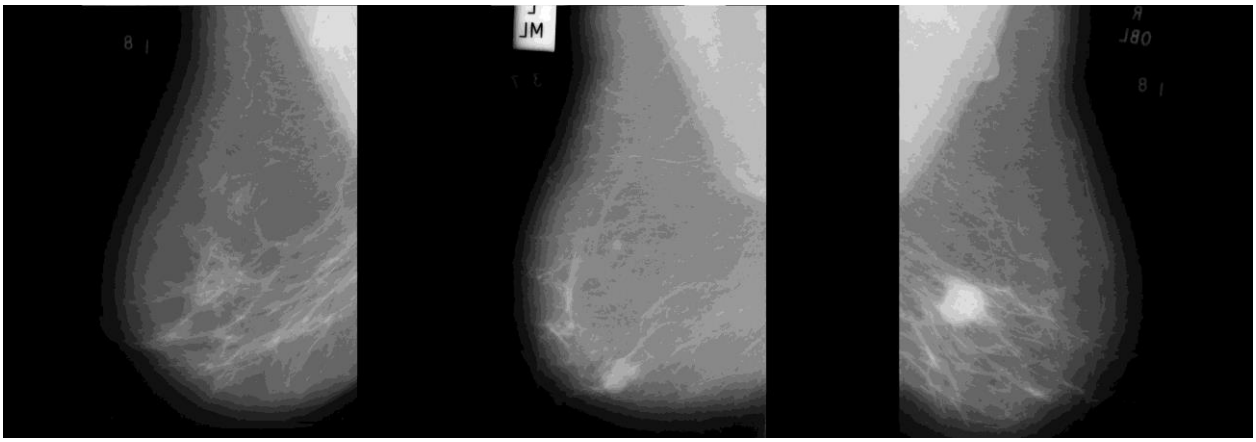

Figure 3

The mammograms analyzed were obtained from the Pilot European Image Processing Archive. The dataset is a subset of an original dataset from the MIAS Database. This subset was utilized since it was easier to obtain and analyze. The original data is saved as a lossless jpeg file type. The file type is cumbersome, and there are few programs readily available to convert the

[6] Fenton, Joshua J., Guibo Xing, Joann G. Elmore, Heejung Bang, Steven L. Chen, Karen K. Lindfors, and Laura-Mae Baldwin. "Short-Term Outcomes of Screening Mammography Using Computer-Aided Detection." *Annals of Internal Medicine Ann Intern Med* 158, no. 8 (2013): 580. doi:10.7326/0003-4819-158-8-201304160-00002.
[7] Imbd.
[8] Bazzocchi, M., F. Mazzarella, C. Del Frate, F. Girometti, and C. Zuiani. "CAD Systems for Mammography: A Real Opportunity? A Review of the Literature." *Radiol Med La Radiologia Medica* 112, no. 3 (2007): 329-53. doi:10.1007/s11547-007-0145-5.

file type. While a formal request was made from another source to obtain JPEG versions of the files, the request was not answered at this time. Thus, this subset was used for the immediate purposes of this paper. This subset has been reduced to 200 micro pixel edge and clipped and/or padded so that every image is 1024 by 1024 pixels. Each image was about one megabyte in size. Each mammogram is black and white, with 256 intensities. There is a total of 322 mammograms in the data set. This means that there are 161 patients. There are 61 breasts with benign tumors, 51 breasts with malignant tumors, and 209 breasts that are normal. In summary, 108 patients have some type of tumor present, while 53 do not. Examples of what breasts that are deemed to be categorized as normal, benign, or cancer are shown in Figure 3 from left to right respectively. The benign case has an abnormality that is oddly shaped and dull in hue. It does not appear to be circular. These are some features that suggest why it is benign. However, the cancer case is clearly circular and bright white. Thus, these features suggest the breast has a malignant tumor.

We then converted the images to the histograms of intensities using ImageJ. We streamlined this process by writing a JavaScript that would make the conversion faster than by hand. We will reference this set of images as the "regular data". While an analysis was performed on the regular data, it was hypothesized that by transforming the images, the desired features presented more apparently and easier to discriminate between normal cases. Each image was transformed by finding the edges using ImageJ by writing a macro to transform the images. Examples of these are provided in Figure 4 of one of the patients. The one on the left is considered normal, while the one on the right has a malignant tumor. Note that the breasts are fairly symmetric, except for a gaping black hole towards the bottom left quadrant. That is the

distinguishing feature that would be used in the Comparison Method. We then took these transformed images and calculated the histograms. We will call these transformed images the 'edges data'.

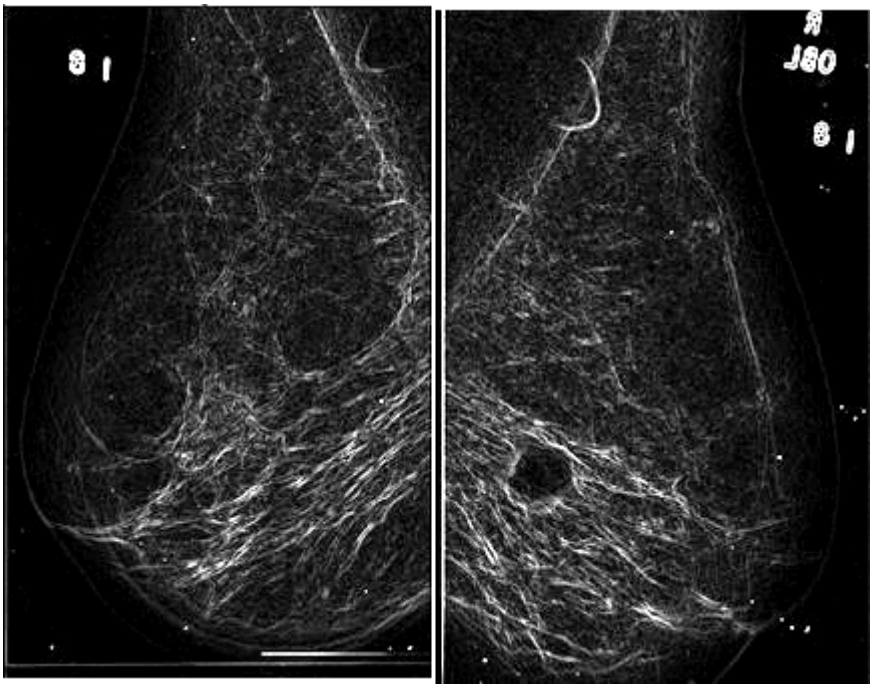

Figure 4 – Note that additional changes were made to these images so that the differences could be seen easier on a variety of different mediums and settings.

The difference between each patient's breasts was taken as to simulate the Comparison Method. This was performed on the regular data and edges data separately. The differences for the regular data will be referenced as 'regular difference data', and the differences for the edges data will be referenced as 'edges difference data'. Thus, there is 4 total data set for analysis. The variables used for the analyses are only the intensities or difference of intensities. Those observations that have a tumor are coded as 0, and those that do not are coded as 1.

Methods

I utilized a variety of methods for the analysis of the 4 data sets. The goal was to be able to discriminate between those with a tumor and those without one. This was the goal as it is similar to the basic goal of CAD, which was to find concerning features in mammograms. While CAD is looking for specific parts of the mammogram, the goal of this analysis strives towards is simply to identify those patients or breasts that have tumors or not. While not all methods will

be presented in this paper, three will be mentioned: support vector machines, principal component analysis, and K-means clustering. The one that performed the best was SVM for all of the data sets. While PCA was about to achieve a very high percentage of the variation explained, none of the principal components seemed able to clearly discriminate between the 2 classes. K-means overall performed very poorly.

Table 1			
Data	**Kernel**	**Parameters**	**Support Vectors**
Regular	Sigmoid	Cost=0.1 Coef0=0.05	163
Edges	Polynomial	Degree=1 Coef0=0.05	181
Regular Difference	Polynomial	Degree=1 Coef0=0.005	82
Edges Difference	Polynomial	Degree=1 Coef0=0.05	85

Table 1 summarizes the best kernels and parameters used in SVM and the number of support vectors for each of the data sets. The data was split into a training and testing data sets and a 10 fold cross validation was performed on each model tested. None of the initial models performed that well with their initial classifications. For example, one of the SVM models classified all of the observations to be normal except for 1 for the edges difference data. However, once we create the ROCs, we can see that the models perform much better if the cutoff probability is adjusted. Figures 5 through 8 are the four ROC curves for each of the models. Note that the red line is for the training data, and the blue line is for the testing data. Figure 9 summarizes Figures 5 through 8.

Figure 9 has the ROC for the edge difference data's training and testing data. However, it also has what I chose to be the optimal points for the other models as well. The purple points with white outlines reference the regular difference data. The green points with a black outline

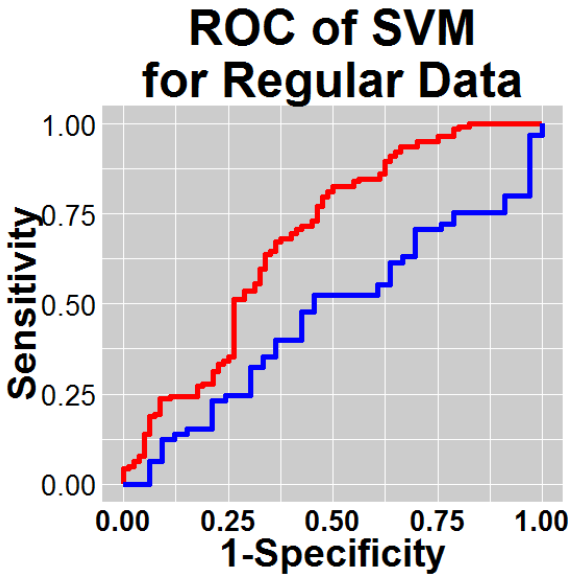

Figure 5

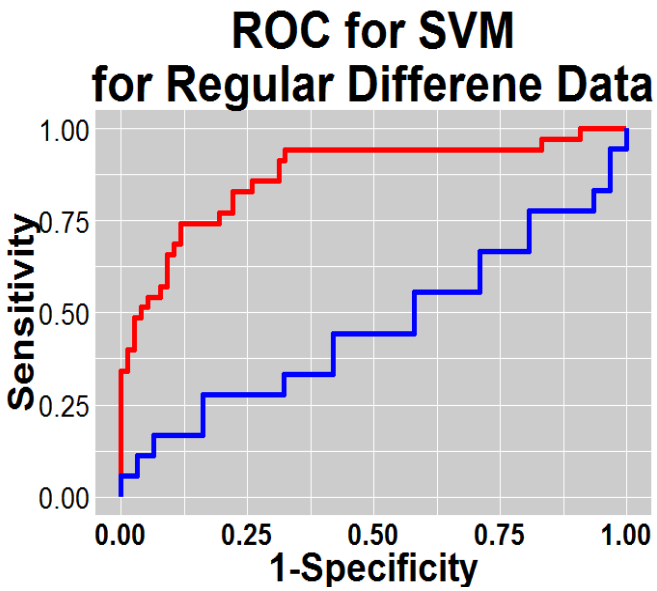

Figure 6

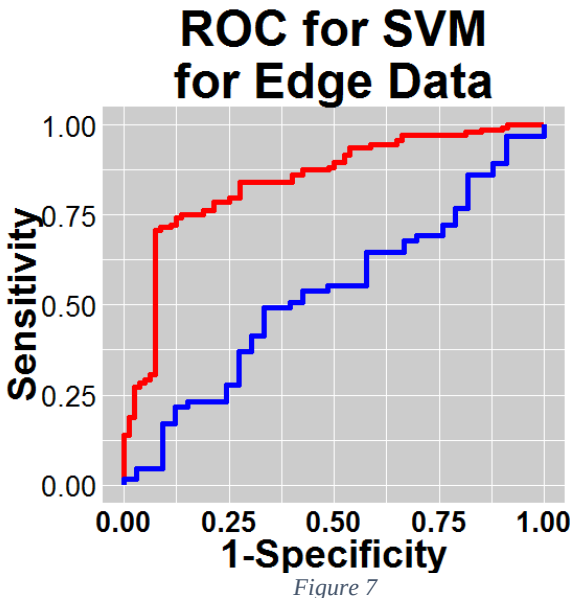

Figure 7

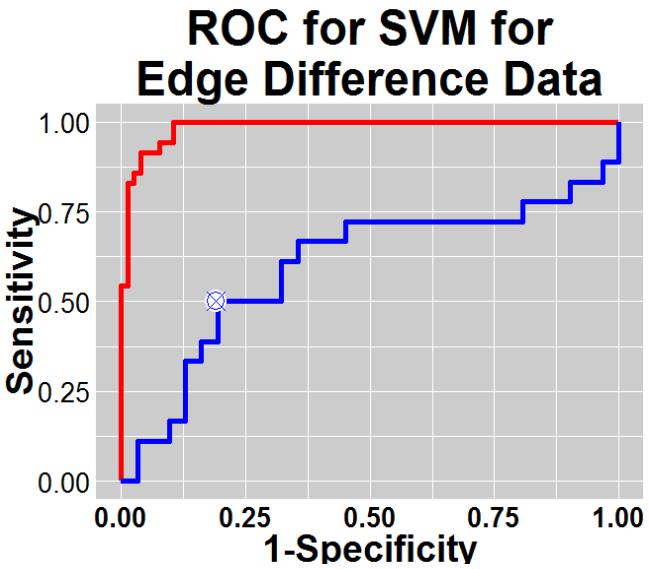

Figure 8

reference the edges data. The cyan with black outline references the regular data. The triangles indicate if it is the training data. When the optimal point was chosen, it was desired that sensitivity, or the proportion of correctly classified normal cases, did not go below 50%. Thus, we note that the edge difference data performed better for the testing and training data sets. Note that while the kernel performed very well on the training data, it did drop significantly on the testing data. However, it was still about to get about 87.5% of the cases where there was a tumor

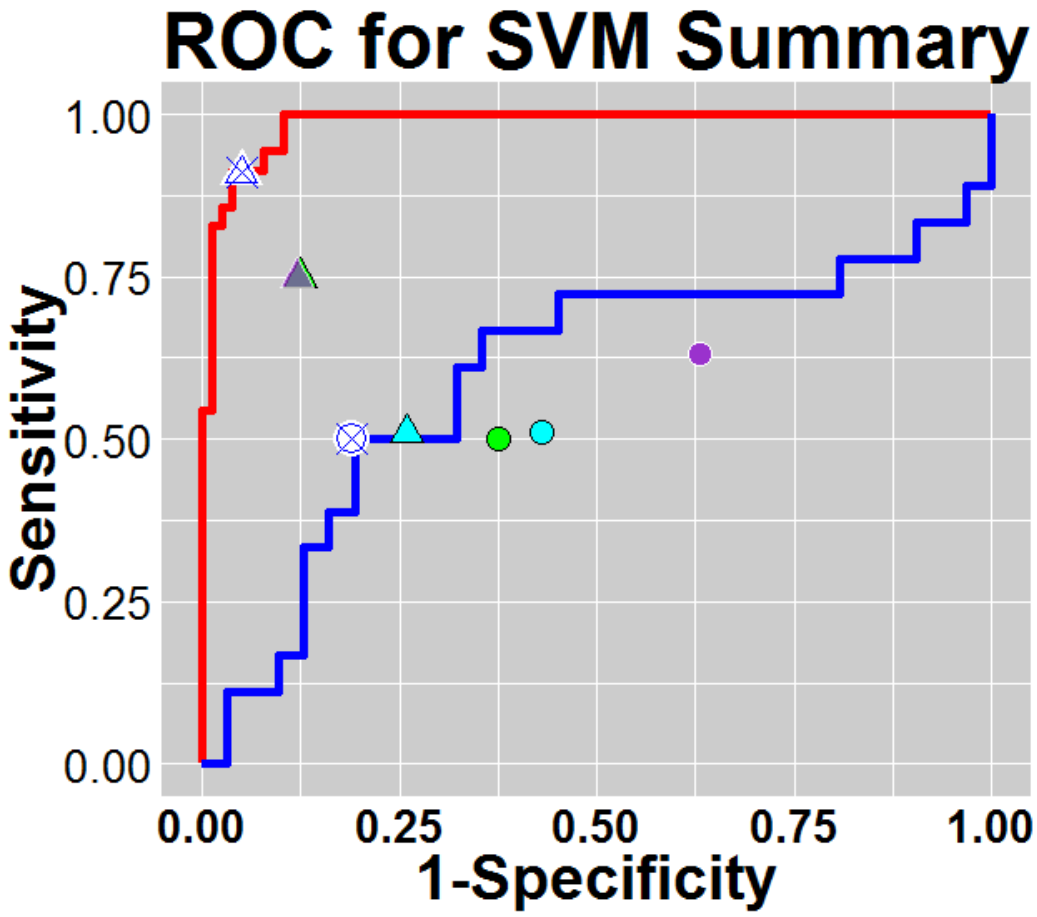

Figure 9

present correctly classified. While the model is only about to get about half of the normal cases correctly, the results are still positive. Given that the model built was relatively simple in nature, this performed very well. For example, this does not consider if the breasts are dense or not. Thus, more complex features can be included at a later time.

For PCA, the results for the regular data and the edges difference data will only be presented. The edges difference data was chosen since it performed the best for SVM, but a brief comparison to the original data is warranted. The other models were omitted to be concise. Upon performing PCA on the edge difference data, note that the proportion of the variation

explained is very high. The first principal component itself explain over 98% of the variation in the data. Plots of the proportion of the variation explained (PVE) by the data are provided in Figure 10. In Figure 11, the PVE for the regular data is presented. Note that while the first principal component was able to explain an overwhelming majority of the data, discriminating features were not able to be found. Plotting a scatterplot matrix of the principal components of the edge difference data was made in an attempt to find these discriminating features, however, due to obtaining so many, they could not all be plotted at once. Thus, plots of various different subsets against one another in an attempt to find those principal components that best discriminated one another. However, none of the ones tried appeared to be able to discriminate well between the two groups. Thus, the first five are presented in Figure 11. Red is used for those that have a tumor. Blue is used for those that are normal. Note however, that the data appears in a linear shape when plotted against the first principal component.

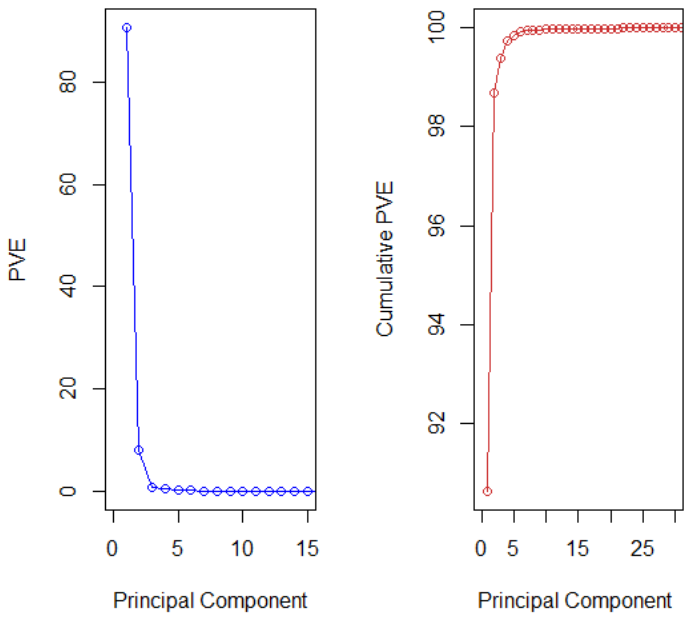

Figure 10

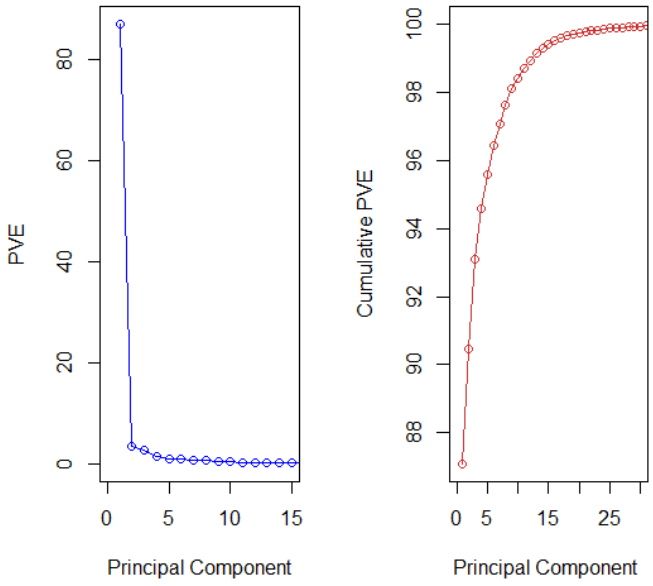

Figure 11

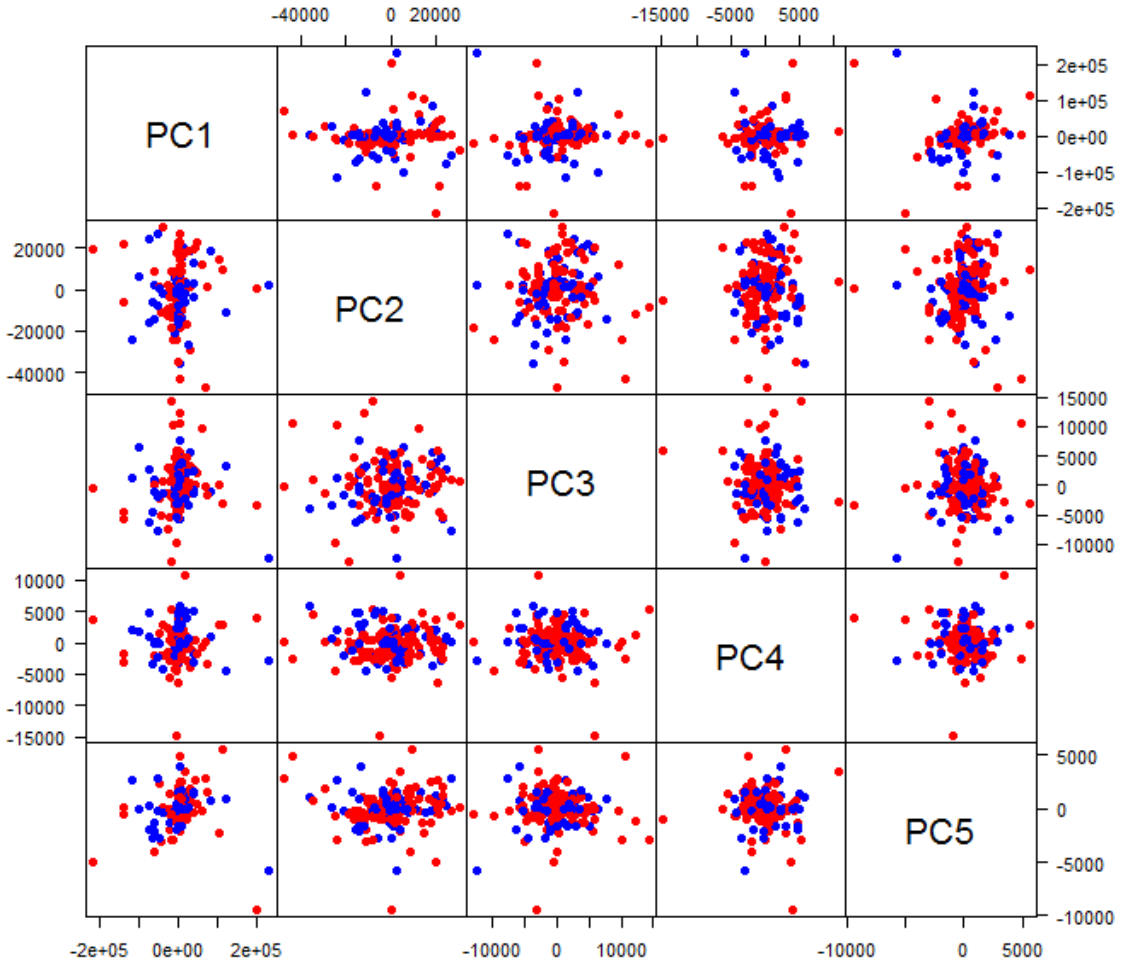

Figure 12

Three dimensional plots were made in an attempt to find discriminating features, but it did not reveal fruitful results either. Note that those that are normal, appear to be massed toward the center. Note that those with a tumor are centered at the core, and then spread out throughout the principal component space.

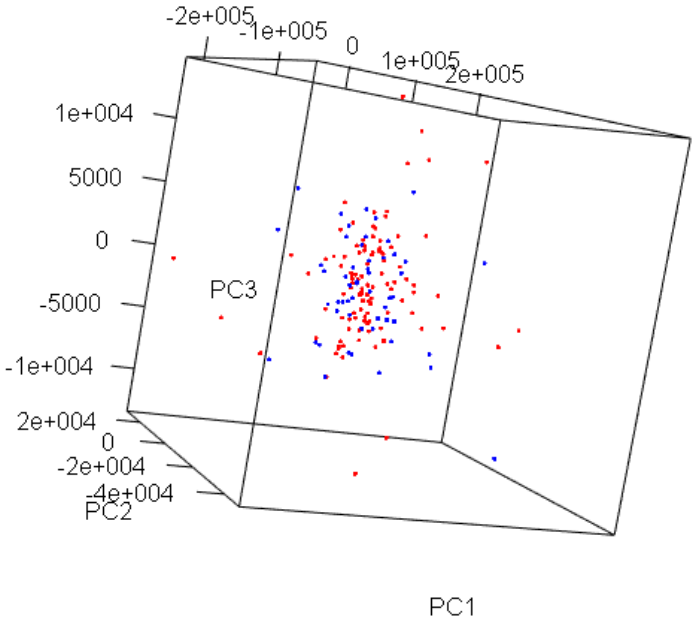

Figure 13

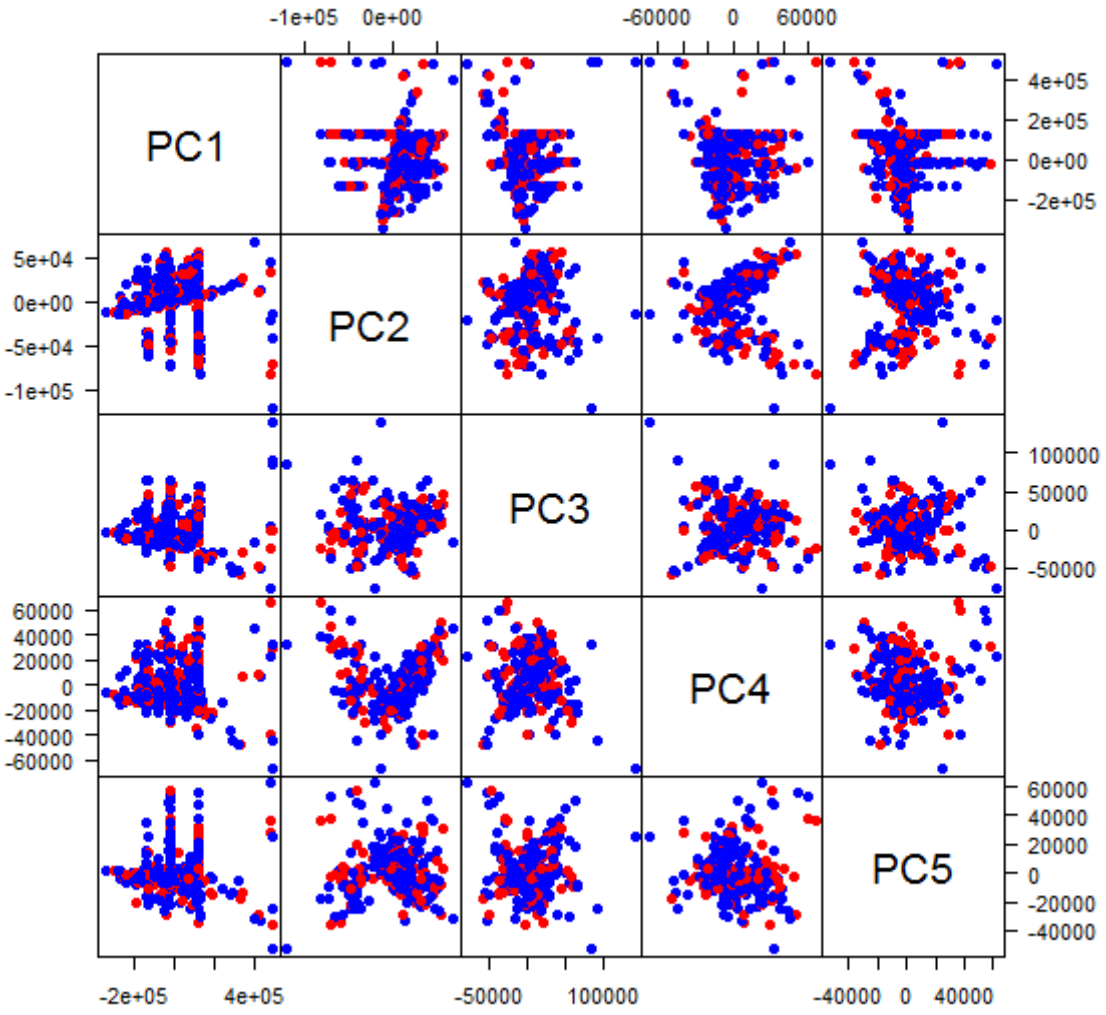

Figure 14

However, this is a very rough description as there are observations scattered throughout. Figure 13 presents a three dimensional plot of the first three principal components with the same coloring scheme as in the previous figure.

A comparison was made with these results to those of the original data. Figure 14 presents a scatterplot matrix for the first five principal components. Again, there were not any prominent features that would help to discriminate between those breasts that have tumors and those that do not. Nonetheless, here are interesting patterns presented in the data. When PC1 is plotted against another, it appears as if there are three distinct lines. One hypothesis is that this

may have to with the fact that we have the left and right breast of each patient. Unfortunately, plots with colors for the right and left breasts did not present any prominent features or characteristics.

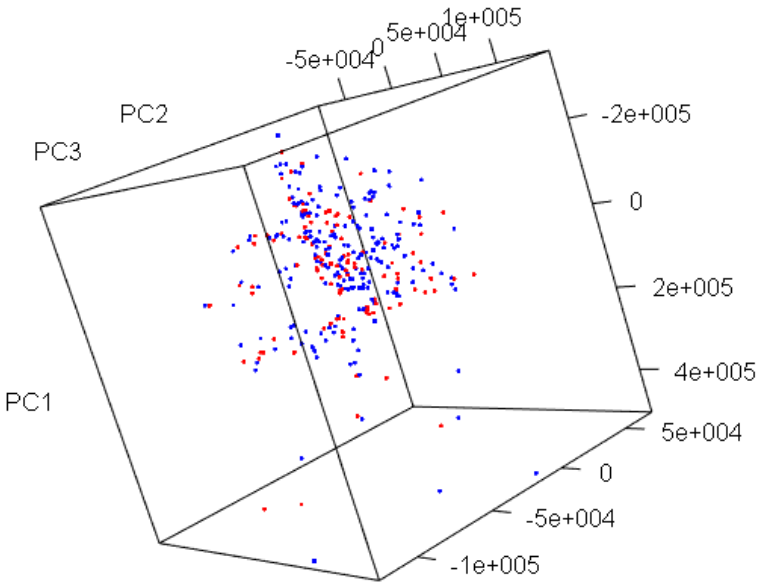

Figure 15

Again, three dimensional plots were made in attempt to find discriminating features in the data, but none were clearly defined. Figure 15 is one of the still images of the first three principal components of this space. As stated previously, there does appear to be three distinct lines in the three dimensional space, but none of the features appear to be able to provide discriminating features for those observations with tumors and for those without tumors. It is not apparent what these patterns in the data are.

For K-means, there will be a focus on the edge difference data, but some of the results from the original data will also be mentioned. Presenting first on the edges data, note that Figure 16, the within sum of squares plot, suggests that two clusters may be promising, but three clusters minimize the within sum or squares better than two clusters. Utilizing both traditional K-means and clustering the data around medoids, three clusters did perform better than two clusters if two of the clusters are combined into one overall cluster. Table 2 shows the comparison between the two methods. Note that the results cluster in essential the same manner.

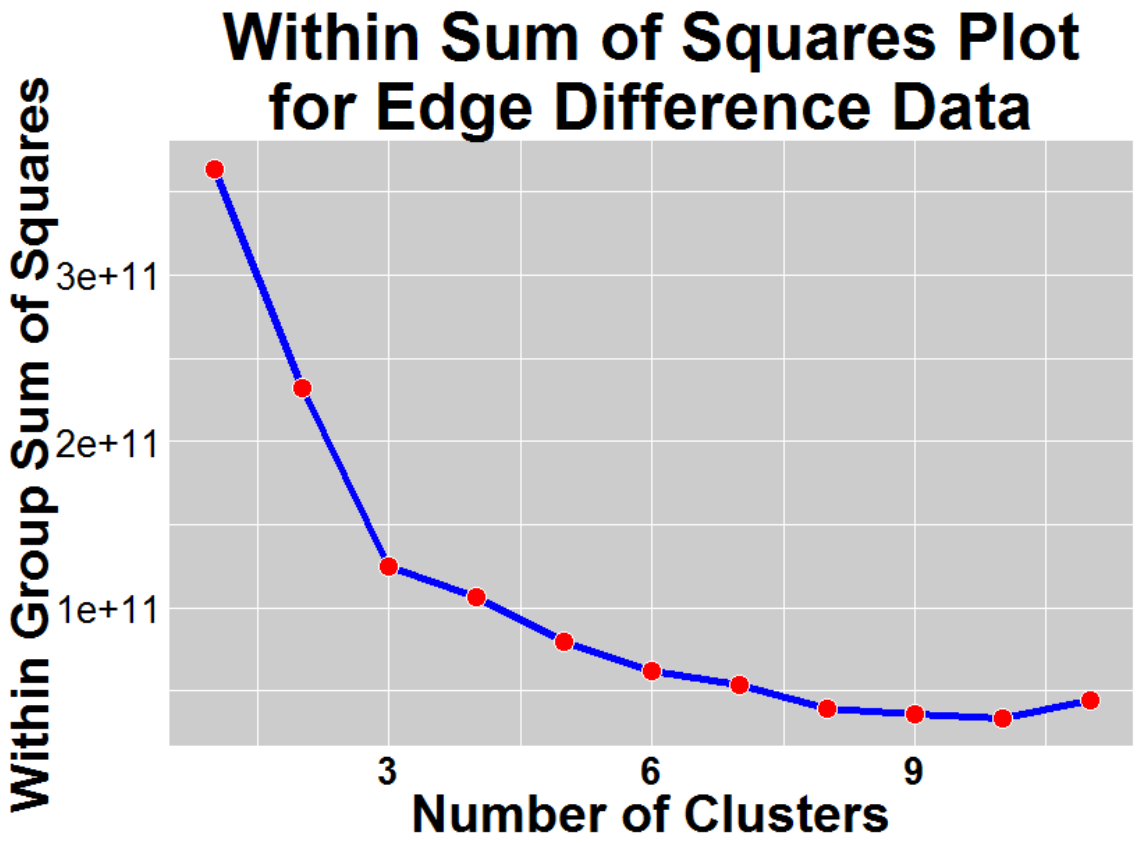

Figure 16

Table 2			
Cluster	**k means:1**	**k means:2**	**k means:3**
Medoids:1	0	0	18
Medoids:2	8	0	0
Medoids:3	1	134	0

Thus, the results for the traditional K-means will only be presented. Table 3 represents the clusters created by k means vs. the actual tumor and normal groups. If the first and third clusters are combined into one overall cluster, which will be called Cluster 2, this will optimize the results. This is summarized in Table 4. Using Table 4, K-means has a correct normal classification rate of 26.4% and a correct tumor classification rate of 89.6%. While this does have a slightly better tumor classification rate than our optimally picked SVM model of about 87%, it has a significant cost for classifying the normal patients. Thus, the SVM model is still the optimal choice.

Table 3

Cluster	Tumor	Normal
k means:1	6	3
k means:2	95	39
k means:3	7	11

Conclusion

The best method for analyzing the data sets given was SVM. Specifically, of all the data sets used, analyzing the difference between patient's transformed mammogram images was the most helpful in discriminating between those patients with a tumor and those without. Overall, the model was able to correctly categorize about 93% of those with tumors and those without in the training data set. In the testing data set, the model was about to correctly categorize about 87% of those with tumors, while only categorizing about 50% of those without tumors.

Discussion

Note that the methods described cannot be directly used to compare the effectiveness of CAD. CAD is used to find concerning areas in the breast. The models presented simplify try to

classify breasts or patients that have normal or concerning features. CAD is used in conjunction with radiologists. CAD will not make decisions on whether or not a feature is even a tumor. It simply presents subsets of an image to the radiologist, and the radiologist must then make a decision on whether to investigate further or not. Our models were not tested with radiologists making real diagnoses.

The hope is that when complete data set is obtained, the models can be refined. By having more data, the number of techniques that would be able to be expanded. For example, performing logistic regression would be much more direct once more observations are observed given that the large number of variables. In regards to the data analyzed in this paper, some improvements to the data itself can be made. As seen in the original examples provided, each mammogram has black segments on the left and right of each image. I would like to be able to remove those to help remove noise from our analysis. However, this would be tricky as while each image are all the same size, that does not guarantee that each mammogram is the same size. The black bands on the right and left of each image do differ in size.

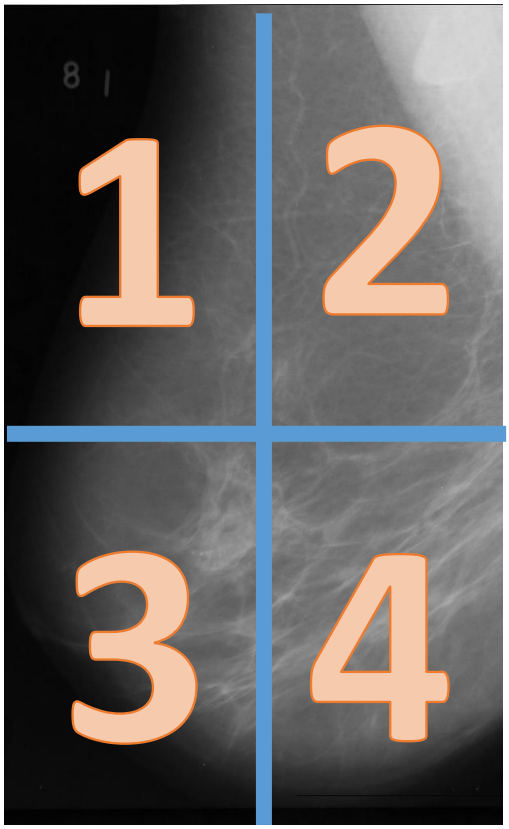

Figure 17

Instead of analyzing the data by transforming it to find the edges within each image, I would also like to create subsets of each image. Then within each subset, we would like to calculate summary statistics like mean and variance. For example, Figure 17 provides an example of what we would desire to do. I would divide the image into parts, and then for

sections 1 through 4, the mean and variance of the parts would be calculated. Thus, these statistics would be used to help differentiate between the mammograms. This in of itself can be tricky as each image is not exactly the same size. Thus, determining the appropriate parts to subdivide the image would require careful considerations. Removing imperfections in the mammograms such as image labels indicating whether the image was of the left or right breast would also be a desired step. Additionally, the images were not recorded by all of the same devices. Being able to account for this in future analyses may also prove helpful.

In addition to the particular aspects of the data itself, including other biological variables may aid progression of future investigations. For example, including the fact that if the patient has a family history of breast cancer, this would be included into the model. Including other categories such as the exact breast densities, amount of fat, existence of previous operations on the breasts, and other important variables should be included. Lastly, building models that can directly distinguish between those with malignant tumors and those who do not is the ultimate goal. Thus, applying the previous recommendations to a model of this nature is the most desirable solution.

Appendix

JavaScript

Macro for finding Edges of Image

```
dir1 = getDirectory("Choose Source Directory ");
dir2 = getDirectory("Choose Destination Directory");

  run("Clear Results");
  setBatchMode(true);
  row = 0;

list = getFileList(dir1);


 for (i=0; i<list.length; i++){
      showProgress(i+1, list.length);
    open(dir1+list[i]);

selectWindow(list[i]);

run("Find Edges");
saveAs("Tiff", dir2+list[i]);

 }
```

Macro for obtaining histograms

```
  dir = getDirectory("Choose a Directory ");
  list = getFileList(dir);
  run("Clear Results");
  setBatchMode(true);
  row = 0;
  for (i=0; i<list.length; i++) {
     showProgress(i, list.length);
     open(dir+list[i]);
     getHistogram(values, counts, 256);
     for(bin=0; bin< values.length; bin++) {
        setResult("Label", row, list[i]);
        setResult("Bin", row, bin+1);
        setResult("Bin Start", row, values[bin]);
        setResult("Count", row, counts[bin]);
        row++;
      }
      close;
  }
  setOption("ShowRowNumbers", false);
  updateResults;
  path = getDirectory("home") + "counts.txt";
  saveAs("Results", path);
```

```
#------------------------------------------------------------------------

R Code

#Cleaning the data

importing data into R for regular files

mydata<-read.csv("Results_Test.csv")

#respahing the data to a usable form

library('reshape')


test.data<-cast(mydata, Label~Bin)

write.csv(test.data, file="Hist Test Data Back 2.csv", row.names=FALSE)


#go into excel and add Norm column

test.data<-read.csv("Hist Test Data.csv")

save(test.data, file="Test Data.RData")

#creating differences data set between pairs of images

load("Test Data.RData")

patient<-rep(c(1:161), 2)
patient<-sort(patient)
patient<-as.factor(patient)


temp<-cbind(patient, test.data)

temp$Label<-NULL
temp$Norm<-NULL

library(plyr)

temp2<-ddply(temp, .(patient), numcolwise(diff))


norm2<-c()

temp.norm<-cbind(patient, test.data$Norm)

for(i in 1:322){

      if(temp.norm[i,1]==temp.norm[i+1,1]){
```

```
            if(temp.norm[i,2]==0 | temp.norm[i+1,2]==0){
            norm2[i]<-0
            }
            else{
            norm2[i]<-1
            }
      }
      else{
      norm2[i]<-NA
      }

}
norm2 <- norm2[!is.na(norm2)]
sum(norm2)

diff.data<-cbind(norm2, temp2)
save(diff.data, file="Diff Test Data.RData")



#SVM
#svm on brest cancer image data
load('Test Data.RData')
test.data$Label<-NULL
data1<-test.data
y<-data1[,1]
data1$Norm<-NULL
y=as.factor(y)
```

```
data1<-cbind(data1, y)

data1=data.frame(data1)

library (e1071)

set.seed (791515)

#training and testing data;
train=sample(321 ,224)

set.seed (791515)

tune.out.poly=tune(svm, y~., data=data1[train ,], kernel ='polynomial',
ranges=list(degree=c( 1, 1.25),ceof0=c(0.005, 1,25, 50)))

summary (tune.out.poly)


#best of the ones we tried....
svmfit =svm(y~., data=data1[train ,], kernel ='polynomial', degree =1,
coef0 =.005)

summary(svmfit)




#taking forever - just skip
tune.out.lin=tune(svm, y~., data=data1[train ,], kernel ='linear',
scale=FALSE,
ranges=list(cost=c(.5, .1,1, 10, 50)))

summary(tune.out.lin)



#
tune.out.sig=tune(svm, y~., data=data1[train ,], kernel ='sigmoid',
ranges=list(cost=c(.5, .1),ceof0=c(0.05, 0.06, 0.07, 0.04)))

summary(tune.out.sig)


svmfit =svm(y~., data=data1[train ,], kernel ='sigmoid', cost =.1, coef0
=.05)

summary(svmfit)

#table of the best model using testing data
table(true=data1[-train,'y'], pred=predict(tune.out.sig$best.model
,newdata=data1[-train,]))
```

```
#ROC Curves

library(ROCR)
library(ggplot2)



mytheme.roc<-theme(

      plot.title = element_text(size=35, face="bold"),
      axis.text.x  = element_text(size=20, face="bold"),
      axis.text.y=element_text(size=20),
      axis.title.x=element_text(size=28, face='bold'),
      axis.title.y=element_text(size=28, face='bold'),
      strip.background=element_rect(fill="gray80"),
      panel.background=element_rect(fill="gray80"),
      axis.ticks= element_blank(),
      axis.text=element_text(colour="black")

      )



#obtaining fitted values
svmfit.opt<-svm(y~., data=data1[train ,], kernel ='polynomial', degree =1,
coef0 =.05, decision.values =T)
fitted=attributes(predict(svmfit.opt ,data1[train ,], decision.values
=TRUE))$decision.values


#produce ROC plot

par(mfrow =c(1,2))


predob=prediction(fitted, data1[train,'y'])
perf=performance(predob, 'tpr', 'fpr')

#convert s4 object to data frame

library('raster')

df.fp<-as.data.frame(perf@x.values)

df.spec<- (1-df.fp)

df.spec<-as.data.frame(df.spec)

colnames(df.spec)<-c('Specificity')

df.sens<-as.data.frame(perf@y.values)

colnames(df.sens)<- c('Sensitivity')
```

```
df.sens<-as.data.frame(df.sens)


svm.roc<-cbind(df.sens, df.spec)

#convert other s4 object -- test data, just as bad

fitted.test=attributes(predict(svmfit.opt ,data1[-train ,],
decision.values =T))$decision.values

predob.test=prediction(fitted.test, data1[-train,'y'])
perf.test=performance(predob.test, 'tpr', 'fpr')

df.fp.test<-as.data.frame(perf.test@x.values)

df.spec.test<- (1-df.fp.test)

df.spec.test<-as.data.frame(df.spec.test)

colnames(df.spec.test)<-c('Specificity')

df.sens.test<-as.data.frame(perf.test@y.values)

colnames(df.sens.test)<- c('Sensitivity')

df.sens.test<-as.data.frame(df.sens.test)

svm.roc.test<-cbind(df.sens.test, df.spec.test)

#roc plot to comapre svm and logistic regression model ---NOTE only run
after logistic model code was run

svm.roc<-as.data.frame(svm.roc)
svm.roc.test<-as.data.frame(svm.roc.test)

svm.roc.1<-matrix(nrow=321, ncol=2)

svm.roc.1[,1]<-c(svm.roc[,1], 4:99)
svm.roc.1[,2]<-c(svm.roc[,2], 4:99)


svm.roc.2<-matrix(nrow=321, ncol=2)
svm.roc.2[,1]<-c(svm.roc.test[,1], 4:225)
svm.roc.2[,2]<-c(svm.roc.test[,2], 4:225)


for(i in 1:321){

      if(svm.roc.1[i,1]>2){

      svm.roc.1[i,1]=NA

      }
```

```
      else{
      svm.roc.1[i,1]=svm.roc.1[i,1]
      }
}

for(i in 1:321){
      if(svm.roc.1[i,2]>2){
      svm.roc.1[i,2]=NA
      }
      else{
      svm.roc.1[i,2]=svm.roc.1[i,2]
      }
}

for(i in 1:321){
      if(svm.roc.2[i,1]>2){
      svm.roc.2[i,1]=NA
      }
      else{
      svm.roc.2[i,1]=svm.roc.2[i,1]
      }
}

for(i in 1:321){
      if(svm.roc.2[i,2]>2){
      svm.roc.2[i,2]=NA
      }
      else{
      svm.roc.2[i,2]=svm.roc.2[i,2]
```

```
      }

}



svm.roc.1<-as.data.frame(svm.roc.1)
svm.roc.2<-as.data.frame(svm.roc.2)



plot.roc.compare<-ggplot(data=svm.roc.1, aes(x=(1-V2),
y=(V1)))+geom_line(colour='red', size=2)+mytheme.roc+
                        xlab("1-Specificity") +
                        ylab("Sensitivity")+
                        ggtitle("ROC of SVM\nfor Regular Data")

#plot.roc.compare<- plot.roc.compare + geom_line(data=NULL, aes(x=1-x.log,
y=y.log), colour='darkgreen')

plot.roc.compare<-plot.roc.compare + geom_line(data=svm.roc.2, aes(x=(1-
V2), y=V1), colour='blue', size=2)


plot.roc.compare



#svm on paired difference brest cancer image data


load('Diff Test Data.RData')

#diff.data<-abs(diff.data)  #observed no differences when we did not use
the absolute value

diff.data$patient<-NULL

data1<-diff.data

y<-data1[,1]

data1$norm2<-NULL

y=as.factor(y)

data1<-cbind(data1, y)

data1=data.frame(data1)

library (e1071)
```

```
set.seed (91490)

#training and testing data;
train=sample(161,112)

set.seed (91490)

tune.out.poly=tune(svm, y~., data=data1[train ,-248], kernel
='polynomial',
ranges=list(degree=c( 1, 1.25),ceof0=c(0.005, 1,25, 50)))

summary (tune.out.poly)


#best of the ones we tried....
svmfit =svm(y~., data=data1[train ,], kernel ='polynomial', degree =1,
coef0 =.005)

summary(svmfit)




#taking forever - just skip
#tune.out.lin=tune(svm, y~., data=data1[train ,], kernel ='linear',
scale=FALSE,
#ranges=list(cost=c(.5, .1,1, 10, 50)))

#summary(tune.out.lin)



#
tune.out.sig=tune(svm, y~., data=data1[train ,-248], kernel ='sigmoid',
#removing X248 since it's essentiall all 0's and cannot scale the value
ranges=list(cost=c(.5, .1),ceof0=c(0.05, 0.06, 0.07, 0.04)), )

summary(tune.out.sig)




#table of the best model using testing data
table(true=data1[-train,'y'], pred=predict(tune.out.poly$best.model
,newdata=data1[-train,]))


#ROC Curves

library(ROCR)
library(ggplot2)
```

```
mytheme.roc<-theme(

      plot.title = element_text(size=35, face="bold"),
      axis.text.x  = element_text(size=20, face="bold"),
      axis.text.y=element_text(size=20),
      axis.title.x=element_text(size=28, face='bold'),
      axis.title.y=element_text(size=28, face='bold'),
      strip.background=element_rect(fill="gray80"),
      panel.background=element_rect(fill="gray80"),
      axis.ticks= element_blank(),
      axis.text=element_text(colour="black")

      )



#obtaining fitted values
svmfit.opt<-svm(y~., data=data1[train ,-248], kernel ='polynomial', degree
=1, coef0 =.005, decision.values =T)
fitted=attributes(predict(svmfit.opt ,data1[train ,], decision.values
=TRUE))$decision.values


#produce ROC plot

par(mfrow =c(1,2))


predob=prediction(fitted, data1[train,'y'])
perf=performance(predob, 'tpr', 'fpr')

#convert s4 object to data frame

library('raster')

df.fp<-as.data.frame(perf@x.values)

df.spec<- (1-df.fp)

df.spec<-as.data.frame(df.spec)

colnames(df.spec)<-c('Specificity')

df.sens<-as.data.frame(perf@y.values)

colnames(df.sens)<- c('Sensitivity')

df.sens<-as.data.frame(df.sens)


svm.roc<-cbind(df.sens, df.spec)

#convert other s4 object -- test data, just as bad
```

```
fitted.test=attributes(predict(svmfit.opt ,data1[-train ,],
decision.values =T))$decision.values

predob.test=prediction(fitted.test, data1[-train,'y'])
perf.test=performance(predob.test, 'tpr', 'fpr')

df.fp.test<-as.data.frame(perf.test@x.values)

df.spec.test<- (1-df.fp.test)

df.spec.test<-as.data.frame(df.spec.test)

colnames(df.spec.test)<-c('Specificity')

df.sens.test<-as.data.frame(perf.test@y.values)

colnames(df.sens.test)<- c('Sensitivity')

df.sens.test<-as.data.frame(df.sens.test)

svm.roc.test<-cbind(df.sens.test, df.spec.test)

#roc plot to comapre svm and logistic regression model ---NOTE only run
after logistic model code was run

svm.roc<-as.data.frame(svm.roc)
svm.roc.test<-as.data.frame(svm.roc.test)

svm.roc.1<-matrix(nrow=161, ncol=2)

svm.roc.1[,1]<-c(svm.roc[,1], 4:51)
svm.roc.1[,2]<-c(svm.roc[,2], 4:51)


svm.roc.2<-matrix(nrow=161, ncol=2)
svm.roc.2[,1]<-c(svm.roc.test[,1], 4:114)
svm.roc.2[,2]<-c(svm.roc.test[,2], 4:114)


for(i in 1:161){

      if(svm.roc.1[i,1]>2){

      svm.roc.1[i,1]=NA

      }

      else{

      svm.roc.1[i,1]=svm.roc.1[i,1]

      }

}
```

```
for(i in 1:161){
      if(svm.roc.1[i,2]>2){
      svm.roc.1[i,2]=NA
      }
      else{
      svm.roc.1[i,2]=svm.roc.1[i,2]
      }
}
for(i in 1:161){
      if(svm.roc.2[i,1]>2){
      svm.roc.2[i,1]=NA
      }
      else{
      svm.roc.2[i,1]=svm.roc.2[i,1]
      }
}

for(i in 1:161){
      if(svm.roc.2[i,2]>2){
      svm.roc.2[i,2]=NA
      }
      else{
      svm.roc.2[i,2]=svm.roc.2[i,2]
      }
}

svm.roc.1<-as.data.frame(svm.roc.1)
svm.roc.2<-as.data.frame(svm.roc.2)
```

```
plot.roc.compare<-ggplot(data=svm.roc.1, aes(x=(1-V2),
y=(V1)))+geom_line(colour='red', size=2)+mytheme.roc+
                        xlab("1-Specificity") +
                        ylab("Sensitivity")+
                        ggtitle("ROC for SVM\nfor Regular Differene Data")

#plot.roc.compare<- plot.roc.compare + geom_line(data=NULL, aes(x=1-x.log,
y=y.log), colour='darkgreen')

plot.roc.compare<-plot.roc.compare + geom_line(data=svm.roc.2, aes(x=(1-
V2), y=V1), colour='blue', size=2)

windows()
plot.roc.compare



#svm on brest cancer image data

load('Test Edge Data.RData')

test.data.edges$Label<-NULL

data1<-test.data.edges

y<-data1[,1]

data1$Norm<-NULL

y=as.factor(y)

data1<-cbind(data1, y)

data1=data.frame(data1)

library (e1071)

set.seed (791515)

#training and testing data;
train=sample(321 ,224)

set.seed (791515)

tune.out.poly=tune(svm, y~., data=data1[train ,], kernel ='polynomial',
ranges=list(degree=c( 1, 1.25),ceof0=c(0.005, 1,25, 50)))

summary (tune.out.poly)


#best of the ones we tried....
```

```
svmfit =svm(y~., data=data1[train ,], kernel ='polynomial', degree =1,
coef0 =.005)

summary(svmfit)


#taking forever - just skip
#tune.out.lin=tune(svm, y~., data=data1[train ,], kernel ='linear',
scale=FALSE,
#ranges=list(cost=c(.5, .1,1, 10, 50)))

#summary(tune.out.lin)



#
tune.out.sig=tune(svm, y~., data=data1[train ,], kernel ='sigmoid',
ranges=list(cost=c(.5, .1),ceof0=c(0.05, 0.06, 0.07, 0.04)))

summary(tune.out.sig)


svmfit =svm(y~., data=data1[train ,], kernel ='sigmoid', cost =.1, coef0
=.05)

summary(svmfit)

#table of the best model using testing data
table(true=data1[-train,'y'], pred=predict(tune.out.sig$best.model
,newdata=data1[-train,]))


#ROC Curves

library(ROCR)
library(ggplot2)


mytheme.roc<-theme(

      plot.title = element_text(size=35, face="bold"),
      axis.text.x  = element_text(size=20, face="bold"),
      axis.text.y=element_text(size=20),
      axis.title.x=element_text(size=28, face='bold'),
      axis.title.y=element_text(size=28, face='bold'),
      strip.background=element_rect(fill="gray80"),
      panel.background=element_rect(fill="gray80"),
      axis.ticks= element_blank(),
      axis.text=element_text(colour="black")

      )
```

```
#obtaining fitted values
svmfit.opt<-svm(y~., data=data1[train ,], kernel ='polynomial', degree =1,
coef0 =.005, decision.values =T)
fitted=attributes(predict(svmfit.opt ,data1[train ,], decision.values
=TRUE))$decision.values


#produce ROC plot

par(mfrow =c(1,2))


predob=prediction(fitted, data1[train,'y'])
perf=performance(predob, 'tpr', 'fpr')

#convert s4 object to data frame

library('raster')

df.fp<-as.data.frame(perf@x.values)

df.spec<- (1-df.fp)

df.spec<-as.data.frame(df.spec)

colnames(df.spec)<-c('Specificity')

df.sens<-as.data.frame(perf@y.values)

colnames(df.sens)<- c('Sensitivity')

df.sens<-as.data.frame(df.sens)


svm.roc<-cbind(df.sens, df.spec)

#convert other s4 object -- test data, just as bad

fitted.test=attributes(predict(svmfit.opt ,data1[-train ,],
decision.values =T))$decision.values

predob.test=prediction(fitted.test, data1[-train,'y'])
perf.test=performance(predob.test, 'tpr', 'fpr')

df.fp.test<-as.data.frame(perf.test@x.values)

df.spec.test<- (1-df.fp.test)

df.spec.test<-as.data.frame(df.spec.test)

colnames(df.spec.test)<-c('Specificity')

df.sens.test<-as.data.frame(perf.test@y.values)
```

```
colnames(df.sens.test)<- c('Sensitivity')

df.sens.test<-as.data.frame(df.sens.test)

svm.roc.test<-cbind(df.sens.test, df.spec.test)

#roc plot to comapre svm and logistic regression model ---NOTE only run
after logistic model code was run

svm.roc<-as.data.frame(svm.roc)
svm.roc.test<-as.data.frame(svm.roc.test)

svm.roc.1<-matrix(nrow=321, ncol=2)

svm.roc.1[,1]<-c(svm.roc[,1], 4:99)
svm.roc.1[,2]<-c(svm.roc[,2], 4:99)


svm.roc.2<-matrix(nrow=321, ncol=2)
svm.roc.2[,1]<-c(svm.roc.test[,1], 4:225)
svm.roc.2[,2]<-c(svm.roc.test[,2], 4:225)


for(i in 1:321){

      if(svm.roc.1[i,1]>2){

      svm.roc.1[i,1]=NA

      }

      else{

      svm.roc.1[i,1]=svm.roc.1[i,1]

      }

}


for(i in 1:321){

      if(svm.roc.1[i,2]>2){

      svm.roc.1[i,2]=NA

      }

      else{

      svm.roc.1[i,2]=svm.roc.1[i,2]

      }
```

```
}

for(i in 1:321){

      if(svm.roc.2[i,1]>2){

      svm.roc.2[i,1]=NA

      }

      else{

      svm.roc.2[i,1]=svm.roc.2[i,1]

      }

}


for(i in 1:321){

      if(svm.roc.2[i,2]>2){

      svm.roc.2[i,2]=NA

      }

      else{

      svm.roc.2[i,2]=svm.roc.2[i,2]

      }

}

svm.roc.1<-as.data.frame(svm.roc.1)
svm.roc.2<-as.data.frame(svm.roc.2)

plot.roc.compare<-ggplot(data=svm.roc.1, aes(x=(1-V2),
y=(V1)))+geom_line(colour='red', size=2)+mytheme.roc+
                        xlab("1-Specificity") +
                        ylab("Sensitivity")+
                        ggtitle("ROC for SVM\nfor Edge Data")

#plot.roc.compare<- plot.roc.compare + geom_line(data=NULL, aes(x=1-x.log,
y=y.log), colour='darkgreen')

plot.roc.compare<-plot.roc.compare + geom_line(data=svm.roc.2, aes(x=(1-
V2), y=V1), colour='blue', size=2)

windows()
plot.roc.compare
```

```
#svm on paired difference brest cancer image data

load('Diff Edge Data.RData')

#diff.data.edges<-abs(diff.data.edges)  #observed no differences when we
did not use the absolute value

diff.data.edges$patient<-NULL

data1<-diff.data.edges

y<-data1[,1]

data1$norm2<-NULL

y=as.factor(y)

data1<-cbind(data1, y)

data1=data.frame(data1)

library (e1071)

set.seed (91490)

#training and testing data;
train=sample(161,112)

set.seed (91490)

tune.out.poly=tune(svm, y~., data=data1[train ,], kernel ='polynomial',
ranges=list(degree=c( 1, 1.25),ceof0=c(0.005, 1,25, 50)))

summary (tune.out.poly)


#best of the ones we tried....
#svmfit =svm(y~., data=data1[train ,], kernel ='polynomial', degree =1,
coef0 =.005)

#summary(svmfit)


#taking forever - just skip
#tune.out.lin=tune(svm, y~., data=data1[train ,], kernel ='linear',
scale=FALSE,
#ranges=list(cost=c(.5, .1,1, 10, 50)))

#summary(tune.out.lin)
```

```
tune.out.sig=tune(svm, y~., data=data1[train ,], kernel ='sigmoid',
ranges=list(cost=c(.5, .1),ceof0=c(0.05, 0.06, 0.07, 0.04)))

summary(tune.out.sig)


#using the best model
svmfit =svm(y~., data=data1[train ,], kernel ='polynomial', degree =1,
coef0 =.005)

summary(svmfit)

#table of the best model using testing data
table(true=data1[-train,'y'], pred=predict(tune.out.sig$best.model
,newdata=data1[-train,]))


#ROC Curves

library(ROCR)
library(ggplot2)



mytheme.roc<-theme(

      plot.title = element_text(size=35, face="bold"),
      axis.text.x  = element_text(size=20, face="bold"),
      axis.text.y=element_text(size=20),
      axis.title.x=element_text(size=28, face='bold'),
      axis.title.y=element_text(size=28, face='bold'),
      strip.background=element_rect(fill="gray80"),
      panel.background=element_rect(fill="gray80"),
      axis.ticks= element_blank(),
      axis.text=element_text(colour="black")

      )

#obtaining fitted values
svmfit.opt<-svm(y~., data=data1[train ,], kernel ='polynomial', degree =1,
coef0 =.05, decision.values =T)
fitted=attributes(predict(svmfit.opt ,data1[train ,], decision.values
=TRUE))$decision.values


#produce ROC plot

par(mfrow =c(1,2))


predob=prediction(fitted, data1[train,'y'])
perf=performance(predob, 'tpr', 'fpr')

#convert s4 object to data frame
```

```
library('raster')

df.fp<-as.data.frame(perf@x.values)

df.spec<- (1-df.fp)

df.spec<-as.data.frame(df.spec)

colnames(df.spec)<-c('Specificity')

df.sens<-as.data.frame(perf@y.values)

colnames(df.sens)<- c('Sensitivity')

df.sens<-as.data.frame(df.sens)


svm.roc<-cbind(df.sens, df.spec)

#convert other s4 object -- test data, just as bad

fitted.test=attributes(predict(svmfit.opt ,data1[-train ,],
decision.values =T))$decision.values

predob.test=prediction(fitted.test, data1[-train,'y'])
perf.test=performance(predob.test, 'tpr', 'fpr')

df.fp.test<-as.data.frame(perf.test@x.values)

df.spec.test<- (1-df.fp.test)

df.spec.test<-as.data.frame(df.spec.test)

colnames(df.spec.test)<-c('Specificity')

df.sens.test<-as.data.frame(perf.test@y.values)

colnames(df.sens.test)<- c('Sensitivity')

df.sens.test<-as.data.frame(df.sens.test)

svm.roc.test<-cbind(df.sens.test, df.spec.test)

#roc plot to comapre svm and logistic regression model ---NOTE only run
after logistic model code was run

svm.roc<-as.data.frame(svm.roc)
svm.roc.test<-as.data.frame(svm.roc.test)

svm.roc.1<-matrix(nrow=161, ncol=2)

svm.roc.1[,1]<-c(svm.roc[,1], 4:51)
svm.roc.1[,2]<-c(svm.roc[,2], 4:51)
```

```
svm.roc.2<-matrix(nrow=161, ncol=2)
svm.roc.2[,1]<-c(svm.roc.test[,1], 4:114)
svm.roc.2[,2]<-c(svm.roc.test[,2], 4:114)


for(i in 1:161){

      if(svm.roc.1[i,1]>2){

      svm.roc.1[i,1]=NA

      }

      else{

      svm.roc.1[i,1]=svm.roc.1[i,1]

      }

}


for(i in 1:161){

      if(svm.roc.1[i,2]>2){

      svm.roc.1[i,2]=NA

      }

      else{

      svm.roc.1[i,2]=svm.roc.1[i,2]

      }

}

for(i in 1:161){

      if(svm.roc.2[i,1]>2){

      svm.roc.2[i,1]=NA

      }

      else{

      svm.roc.2[i,1]=svm.roc.2[i,1]

      }
```

```
}


for(i in 1:161){

      if(svm.roc.2[i,2]>2){

      svm.roc.2[i,2]=NA

      }

      else{

      svm.roc.2[i,2]=svm.roc.2[i,2]

      }

}

svm.roc.1<-as.data.frame(svm.roc.1)
svm.roc.2<-as.data.frame(svm.roc.2)

#point for best
x.points<-rep(NA, 161)
y.points<-rep(NA, 161)

x.points[1]<-.19
y.points[1]<-.50

#points for paired regular
x.points.diff.tri<-rep(NA, 161)
y.points.diff.tri<-rep(NA, 161)

x.points.diff.tri[1]<-.12
y.points.diff.tri[1]<-.75

#---------------------------
x.points.diff<-rep(NA, 161)
y.points.diff<-rep(NA, 161)

x.points.diff[2]<-.63
y.points.diff[2]<-.63


#points for edges
x.points.edges.tri<-rep(NA, 161)
y.points.edges.tri<-rep(NA, 161)

x.points.edges.tri[1]<-.125
y.points.edges.tri[1]<-.75


#---------------------------
x.points.edges<-rep(NA, 161)
```

```
y.points.edges<-rep(NA, 161)

x.points.edges[2]<-.375
y.points.edges[2]<-.5


#points for regular
x.points.reg.tri<-rep(NA, 161)
y.points.reg.tri<-rep(NA, 161)

x.points.reg.tri[1]<-.26
y.points.reg.tri[1]<-.51

#---------------------------

x.points.reg<-rep(NA, 161)
y.points.reg<-rep(NA, 161)

x.points.reg[2]<-.43
y.points.reg[2]<-.51

plot.roc.compare<-ggplot(data=svm.roc.1, aes(x=(1-V2),
y=(V1)))+geom_line(colour='red', size=2)+mytheme.roc+
                        xlab("1-Specificity") +
                        ylab("Sensitivity")+
                        ggtitle("ROC for SVM Summary")

plot.roc.compare<-plot.roc.compare + geom_line(data=svm.roc.2, aes(x=(1-
V2), y=V1), colour='blue', size=2)+
                        geom_point(data=NULL, aes(x=x.points, y=y.points),
shape= 21, colour="white", fill='white', size=7.5)+
                        geom_point(data=NULL, aes(x=x.points, y=y.points),
shape= 13, colour="blue", fill='white', size=6)+

                        geom_point(data=NULL, aes(x=x.points.edges.tri,
y=y.points.edges.tri), shape= 24, colour="black", fill='green', size=5)+
                        geom_point(data=NULL, aes(x=x.points.diff.tri,
y=y.points.diff.tri), shape= 24, colour="white", fill='darkorchid3',
size=5)+
                        geom_point(data=NULL, aes(x=x.points.reg.tri,
y=y.points.reg.tri), shape= 24, colour="black", fill='cyan', size=5)+


                        geom_point(data=NULL, aes(x=x.points.edges,
y=y.points.edges), shape= 21, colour="black", fill='green', size=5)+
                        geom_point(data=NULL, aes(x=x.points.diff,
y=y.points.diff), shape= 21, colour="white", fill='darkorchid3', size=5)+
                        geom_point(data=NULL, aes(x=x.points.reg,
y=y.points.reg), shape= 21, colour="black", fill='cyan', size=5)



windows()
plot.roc.compare
```

```
pairs(pr.out$x[,1:14],gap=0,las=1,pch=19,cex.main=.9,
  bg=test.data$norm2)


#PCA


#performing PCA on the breast cancer data


load('Test Data.RData')

labs<-read.csv("Test data labs.csv")

labs1<-test.data$Norm


test.data$Label<-NULL


data1<-test.data

y<-data1[,1]

data1$Norm<-NULL

pr.out=prcomp(data1 , scale=FALSE)

pr.out$rotation[,1]



Cols=function(vec)
{

      cols=rainbow(length(unique(vec)))
      return(cols[as.numeric (as.factor(vec))])

}



par(mfrow=c(2,2))

plot(pr.out$x[,1:2], col=Cols(labs1), pch=19, xlab="Z1",ylab="Z2")

plot(pr.out$x[,c(1,3)], col=Cols(labs1), pch=19, xlab="Z1",ylab="Z3")

plot(pr.out$x[,c(1,4)], col=Cols(labs1), pch=19, xlab="Z1",ylab="Z4")

plot(pr.out$x[,c(1,5)], col=Cols(labs1), pch=19, xlab="Z1",ylab="Z5")
```

```
library(rgl)


pcList <-pr.out

pc <- pcList$x
pcMeans <- colMeans(pc)
round(pcMeans,2)

pcCov <- var(pc)
round(pcCov,2)

summary(pr.out)

plot(pr.out)


fixLights <- function(specular=gray(c(.3,.3,0))){
  clear3d(type="lights")
  light3d(theta=-50,phi=40,
    viewpoint.rel=TRUE, ambient=gray(.7),
    diffuse=gray(.7),specular=specular[1])

  light3d(theta=50,phi=40,
    viewpoint.rel=TRUE, ambient=gray(.7),
    diffuse=gray(.7),specular=specular[2])

  light3d(theta=0,phi=-70,
    viewpoint.rel=TRUE, ambient=gray(.7),
    diffuse=gray(.7),specular=specular[3])
}


data1$Labs<-labs1

temp<-cbind(y, pr.out$x)


cols <- character(nrow(temp))
cols[] <- "black"

cols[temp[,1] == c("1")] <- "blue"
cols[temp[,1] == "0"] <- "red"


pairs(temp[,c(2,16:21)],gap=0,las=1,pch=19,cex.main=.9,
  col=cols)


colors<-c("red","blue")
```

```
data1$Labs<-labs1

#data1$Labs <- factor(labs1, levels = c("B","M","N","?"))


get_colors <- function(groups, group.col = palette()){ #from
http://www.sthda.com/english/wiki/a-complete-guide-to-3d-visualization-
device-system-in-r-r-software-and-data-visualization
  groups <- as.factor(groups)
  ngrps <- length(levels(groups))
  if(ngrps > length(group.col))
    group.col <- rep(group.col, ngrps)
  color <- group.col[as.numeric(groups)]
  names(color) <- as.vector(groups)
  return(color)
}


data1$Left<-rep(c(1,0),161)

open3d()
fixLights()
plot3d(pc[,1:3], box=TRUE,
xlab="PC1",ylab="PC2",zlab="PC3", col=get_colors(data1$Labs, colors)) #red
is cancer       blue is normal

snapshot3d(file="First 3 PC ver 2.png")


#performing PCA on the brest cancer data


load('Diff Edge Data.RData')

#labs<-read.csv("Test data labs.csv")

test.data<-diff.data.edges

labs1<-test.data$norm2


test.data$patient<-NULL


data1<-test.data

y<-data1[,1]

data1$norm2<-NULL

pr.out=prcomp(data1 , scale=FALSE)

pr.out$rotation[,1]
```

```
Cols=function(vec)
{

      cols=rainbow(length(unique(vec)))
      return(cols[as.numeric (as.factor(vec))])

}


par(mfrow=c(2,2))

plot(pr.out$x[,1:2], col=Cols(labs1), pch=19, xlab="PC1",ylab="PC2") #red
is cancer; cyan is normal

plot(pr.out$x[,c(1,3)], col=Cols(labs1), pch=19, xlab="PC1",ylab="PC3")

plot(pr.out$x[,c(1,4)], col=Cols(labs1), pch=19, xlab="PC1",ylab="PC4")

plot(pr.out$x[,c(1,5)], col=Cols(labs1), pch=19, xlab="PC1",ylab="PC5")

temp<-cbind(y, pr.out$x)


cols <- character(nrow(temp))
cols[] <- "black"

cols[temp[,1] == c("1")] <- "blue"
cols[temp[,1] == "0"] <- "red"


pairs(temp[,c(2:6)],gap=0,las=1,pch=19,cex.main=.9,
  col=cols)


library(rgl)


pcList <-pr.out

pc <- pcList$x
pcMeans <- colMeans(pc)
round(pcMeans,2)

pcCov <- var(pc)
#round(pcCov,2)

summary(pr.out)

plot(pr.out)


fixLights <- function(specular=gray(c(.3,.3,0))){
```

```
  clear3d(type="lights")
  light3d(theta=-50,phi=40,
    viewpoint.rel=TRUE, ambient=gray(.7),
    diffuse=gray(.7),specular=specular[1])

  light3d(theta=50,phi=40,
    viewpoint.rel=TRUE, ambient=gray(.7),
    diffuse=gray(.7),specular=specular[2])

  light3d(theta=0,phi=-70,
    viewpoint.rel=TRUE, ambient=gray(.7),
    diffuse=gray(.7),specular=specular[3])
}

colors<-c("red","blue")

data1$Labs<-labs1

#data1$Labs <- factor(labs1, levels = c("B","M","N","?"))

get_colors <- function(groups, group.col = palette()){ #from
http://www.sthda.com/english/wiki/a-complete-guide-to-3d-visualization-
device-system-in-r-r-software-and-data-visualization
  groups <- as.factor(groups)
  ngrps <- length(levels(groups))
  if(ngrps > length(group.col))
    group.col <- rep(group.col, ngrps)
  color <- group.col[as.numeric(groups)]
  names(color) <- as.vector(groups)
  return(color)
}

open3d()
fixLights()
plot3d(pc[,1:3], box=TRUE,
xlab="PC1",ylab="PC2",zlab="PC3", col=get_colors(data1$Labs, colors)) #red
is cancer       blue is normal

snapshot3d(file="First 3 PC.png")


#K-means

#k means-----------------------------------------------------------------

library(useful)

load('Test Data.RData')

test.data<-test.data[,-1]

wssplot <- function(data, nc=15, seed=791515){
               wss <- (nrow(data)-1)*sum(apply(data,2,var))
               for (i in 2:nc){
```

```
                    set.seed(seed)
                    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
                plot(1:nc, wss, type="b", xlab="Number of Clusters",
                     ylab="Within groups sum of squares")}


mydata<-test.data
mydata$Norm<-NULL

eucDist <- dist(mydata)

wssplot(mydata) #says 4 clusters best, but maybe 2?

km2 <- kmeans(mydata, centers=2, nstart = 20)

geneBest <- FitKMeans(mydata, max.clusters=20,nstart=25,seed=43)
geneBest
PlotHartigan(geneBest)

library(MASS)

eucDistAll <- dist(mydata)

library('cluster')

pam2 <- pam(eucDistAll, k=2, diss=TRUE)
pam3 <- pam(eucDistAll, k=3, diss=TRUE)
pam6 <- pam(eucDistAll, k=6, diss=TRUE)

pam2

# See if the gene K-means and pam class labels are the same
# for 2 clusters

all(names(km2$cluster) == names(pam2$clustering))
table(km2$cluster, pam2$clustering)

library(RColorBrewer)

three.class<-read.csv("3 Class.csv")

t.c<-as.matrix(three.class)

table(km2$cluster, test.data[,1])

table(pam2$clustering, test.data[,1])

table(pam3$clustering, t.c)



#k means for diff edges-----------------------------------------------------

library(useful)
```

```
load('Diff Edge Data.RData')

test.data.edges<-(diff.data.edges[,-2])

wssplot <- function(data, nc=15, seed=791515){
               wss <- (nrow(data)-1)*sum(apply(data,2,var))
               for (i in 2:nc){
                    set.seed(seed)
                    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
                plot(1:nc, wss, type="b", xlab="Number of Clusters",
                     ylab="Within groups sum of squares",
                             main="Within Sum of Squares Plot")}

mydata<-test.data.edges
mydata$norm2<-NULL

eucDist <- dist(mydata)

windows()
wssplot(mydata) #says 4 clusters best, but maybe 2?


#creating wss plot to decide how many cluster; we will use 2 and 3 as it
has the steepest drops
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
for (i in 2:11){

       set.seed(791515)
     wss[i] <- sum(kmeans(mydata, centers=i)$withinss)

       }
library(ggplot2)

mytheme.roc<-theme(

      plot.title = element_text(size=35, face="bold"),
      axis.text.x  = element_text(size=20, face="bold"),
      axis.text.y=element_text(size=20),
      axis.title.x=element_text(size=28, face='bold'),
      axis.title.y=element_text(size=28, face='bold'),
      strip.background=element_rect(fill="gray80"),
      panel.background=element_rect(fill="gray80"),
      axis.ticks= element_blank(),
      axis.text=element_text(colour="black")

      )


W<-ggplot(data=NULL, aes(x=1:11, y=wss))+geom_line(color='blue', size=2)+
   geom_point(data=NULL, aes(x=1:11, y=wss), shape= 21, colour="white",
fill='red', size=5)+
   xlab("Number of Clusters") +
   ylab("Within Group Sum of Squares")+
```

```
   ggtitle("Within Sum of Squares Plot\nfor Edge Difference Data")+
   mytheme.roc

windows()
W

km2 <- kmeans(mydata, centers=2, nstart = 20)

km3 <- kmeans(mydata, centers=3, nstart = 20)

geneBest <- FitKMeans(mydata, max.clusters=20,nstart=25,seed=43)
geneBest
#windows()
#PlotHartigan(geneBest)


library(MASS)

eucDistAll <- dist(mydata)

library('cluster')

pam2 <- pam(eucDistAll, k=2, diss=TRUE)
pam3 <- pam(eucDistAll, k=3, diss=TRUE)
#pam6 <- pam(eucDistAll, k=6, diss=TRUE)

pam2

# See if the gene K-means and pam class labels are the same
# for 2 clusters

all(names(km2$cluster) == names(pam2$clustering))
table(km2$cluster, pam2$clustering)



all(names(km3$cluster) == names(pam3$clustering))
table(km3$cluster, pam3$clustering)

three.class<-read.csv("3 Class.csv")

t.c<-as.matrix(three.class)

table(km2$cluster, diff.data.edges[,1])

table(pam2$clustering, diff.data.edges[,1])

table(km3$cluster, diff.data.edges[,1])

table(pam3$clustering, diff.data.edges[,1])
```