

Meow Classification

Jason Kang

6/9/2019

A. Intro

I will be predicting what cats are trying to say based on self-interpreted labels using support vector machine. "Cats meow when they are trying to say something to humans" (Psychology Today, 5 Sep. 2018). Although I'm not a cat linguist, I have two cats at home for many years and have come to understand what my cats are trying to say.

Using the audio files of cats from kaggle (<https://www.kaggle.com/mmoreaux/audio-cats-and-dogs/version/5>), I've sampled and labeled 77 audio files and split train and test data by half. Cats are mysterious creatures so we will see how that affects our accuracy. This is merely an attempt to systematically translating cat communication.

B. Methodology

As mentioned, we will be using a multi-class support vector machine as there are many factors to consider when determining the outcome. We will be calling the following libraries to implement from reading the wav files all the way to prediction.

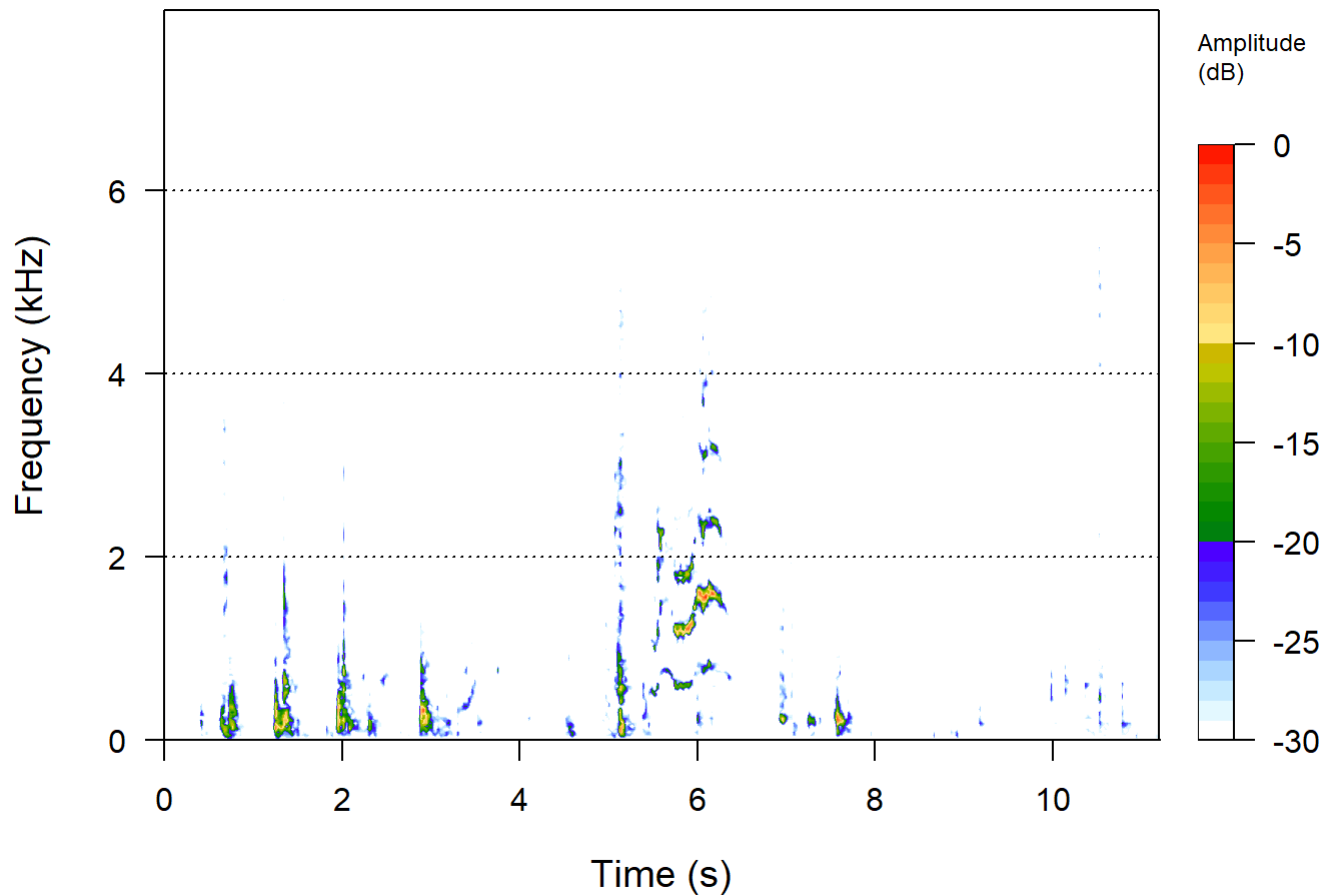
TuneR, seewave, psych, clusterSim, nFactgors, caret, e1071, class, dplyr

```
library(tuneR)
library(seewave)
library(psych)
library(clusterSim)
library(nFactors)
library(caret)
library(e1071)
library(class)
library(dplyr)
```

C. Data Exploration and Visualization

Next, we want to load the path, extrapolate descriptive statistics from the wav files, take only the 'left' piece of information.

```
filenames <- list.files(path = "C:/Users/Kangs/Documents/cataudio/catstrain/", pattern = "wav",
  full.names = TRUE) #load path of files
d<- lapply(filenames, readWave) #read all the wav files into d
names(d) <- gsub(".*/(.*)\\.\\..*", "\\1", filenames) # use file name as vector name
left<-sapply(d, function(x) x@left) # take left of wav file descriptive statistics
spectro(d$cat_1) #this is a spectrogram and creating numeric value from this will help create useable attributes for the model
```



D. Data Cleaning and Labeling

Then I labeled the variables after tabularizing with statistical measures (mean, sd, skew, kurtosis, etc). These might be good indicators for the algorithm of what the cat is trying to say. What I did was listened to each audio file and labeled them here audio by audio. In this example, the cat may have said "Stop" when meowing. I'm just showing 1 of 77 audio file label method here.

```
x<- left$cat_1 #vector
x1<- 1 #file number
amplitude <- Mod(fft(x)[1:round(length(fft(x))/2,0)]) #get amplitude for each data point
featuredata<-cbind(describe(x)[,3:13],describe(amplitude)[,3:13],"Stop", x1) #extract statistical measures
write.table(featuredata,"C:/Users/Kangs/Documents/cataudio/catstrain/test/cat_audio_nonfinal.csv",quote = FALSE, sep = ",", row.names = FALSE, col.names = TRUE, append = FALSE ) #store into csv
```

E. Implementation

After gathering the labels, I create the feature table and split the dataset in half for test and train groups

```
#import feature data table
featureData<-read.csv("C:/Users/Kangs/Documents/cataudio/catstrain/test/cat_audio_data.csv", header = TRUE, sep = ",")
ft<-cbind(featureData[,1:22],featureData$label)
smp_size <- floor(0.50 * nrow(ft)) # 50% split of feature table

set.seed(123) ## set the seed to make your partition reproducible
train_ind <- sample(seq_len(nrow(ft)), size = smp_size)

train <- scale(ft[train_ind,1:22 ], center=TRUE)
test <- scale(ft[-train_ind,1:22 ], center=TRUE)

trainlabel<-ft[train_ind,23 ]
testlabel<-ft[-train_ind,23 ]
```

F. Modeling and Accuracy

Here we create the model and predict it on the test set

```
model_svm <- svm(trainlabel ~ train ) #SVM model

pred <- predict(model_svm, test) #Use the predictions on the test dataset
confusionMatrix(pred,testlabel)
```

Confusion Matrix and Statistics

```
##
##               Reference
## Prediction    Hello? I'm hungry I'm Hungry I love you I missed you
## Hello?        8      4      2      2      2
## I'm hungry     0      0      0      0      0
## I'm Hungry     0      0      0      0      0
## I love you     0      0      0      0      0
## I missed you   1      0      0      0      0
## Play with me   0      0      0      0      0
## Shh            0      0      1      0      0
## Stop           0      0      0      0      0
## What's happening? 0      0      0      0      0
```

```
##               Reference
## Prediction    Play with me Shh Stop What's happening?
## Hello?        5      0      2      4
## I'm hungry     0      1      1      1
## I'm Hungry     0      0      0      0
## I love you     2      0      0      0
## I missed you   0      0      0      0
## Play with me   0      0      0      0
## Shh            1      0      0      1
## Stop           0      0      0      0
## What's happening? 0      0      0      0
```

##

Overall Statistics

```
##
##               Accuracy : 0.2105
##               95% CI : (0.0955, 0.3732)
##       No Information Rate : 0.2368
##       P-Value [Acc > NIR] : 0.7082
##
##               Kappa : 0.0189
##
## Mcnemar's Test P-Value : NA
```

Statistics by Class:

```
##
##               Class: Hello? Class: I'm hungry Class: I'm Hungry
## Sensitivity      0.8889      0.00000      0.00000
## Specificity      0.2759      0.91176      1.00000
## Pos Pred Value    0.2759      0.00000      NaN
## Neg Pred Value    0.8889      0.88571      0.92105
## Prevalence        0.2368      0.10526      0.07895
## Detection Rate    0.2105      0.00000      0.00000
## Detection Prevalence 0.7632      0.07895      0.00000
## Balanced Accuracy 0.5824      0.45588      0.50000
##
##               Class: I love you Class: I missed you
## Sensitivity      0.00000      0.00000
## Specificity      0.94444      0.97222
## Pos Pred Value    0.00000      0.00000
## Neg Pred Value    0.94444      0.94595
## Prevalence        0.05263      0.05263
```

```

## Detection Rate          0.00000      0.00000
## Detection Prevalence    0.05263      0.02632
## Balanced Accuracy       0.47222      0.48611
##
##          Class: Play with me Class: Shh Class: Stop
## Sensitivity              0.0000      0.00000      0.00000
## Specificity              1.0000      0.91892      1.00000
## Pos Pred Value           NaN          0.00000          NaN
## Neg Pred Value           0.7895      0.97143      0.92105
## Prevalence               0.2105      0.02632      0.07895
## Detection Rate           0.0000      0.00000      0.00000
## Detection Prevalence     0.0000      0.07895      0.00000
## Balanced Accuracy        0.5000      0.45946      0.50000
##
##          Class: What's happening?
## Sensitivity              0.0000
## Specificity              1.0000
## Pos Pred Value           NaN
## Neg Pred Value           0.8421
## Prevalence               0.1579
## Detection Rate           0.0000
## Detection Prevalence     0.0000
## Balanced Accuracy        0.5000

```

G. Results and Conclusion

Cats are extremely mysterious. And as more time permits, there is huge opportunity to creating a more robust model perhaps utilizing random forest may increase the accuracy along with hiring a cat linguistic expert to labeling each audio and increasing the sample of labels. Another method to tinker with is doing image classification of the spectrogram. Overall, I would say this is a good start to developing a product that may be in demand for many consumers/cat owners.