[Tyler Castelli

ECE 3522: Stochastic Processes in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 1912

# Problem Statement

The goal of this assignment was to fit a distribution to each of the data sets given to us. First we fit a Gaussian distribution to each set, then were required to find an alternate distribution to the Google Stock data set.

# Approach and Results

For this assignment I created histograms of each set of data and normalized them to create values for a pdf of each data set. In order to capture all of the data in a bin, I used a bin size of 25 for the Google Stock and a bin size of 5 for the audio signal. Them, using the middle values of each bin, I fit a Gaussian distribution over each data set, giving the plots below.

Figure 1

Figure 2

We can clearly see that the Gaussian distribution for the audio signal is a relatively nice fit, however for the Google stock it is a rather poor fit. I then used both a Poisson and an Erlang distribution and attempted to fit these over the Google Stock values. The results are shown below in figure 3.


Figure 3

Again, we see that none of these distributions is a very good fit for the Google stock data. we will discuss this in the conclusion section.

# MATLAB Code

%%

%Part 1

clear all; close all; clc;

%Open file

f1 = fopen('rec\_01\_speech.raw','r');

a\_sig = fread(f1,inf,'int16');

fclose(f1);

google\_data = xlsread('google\_v00');

g\_stock = google\_data(:,4);

%g\_stock = [1 2 3 4 5 2 3 3 3 4 4 4 3 8 1 2 7 4 3 3 5 2 3 2 3 2 3 3 3 3 4 3 1 0 5 7 4 5 7 8 7 7 8 2 3 4 4 5 5];

g\_bin = 25;

a\_bin = 5;

g\_binstart = min(g\_stock);

g\_binlimit = max(g\_stock);

p1 = figure(1);

p2 = figure(2);

%Plot the histogram of the Google stock

figure(1)

subplot(1,1,1);

hold on;

h1 = histogram(g\_stock,'BinWidth',g\_bin); %change width of the bins

h1.BinLimits = [g\_binstart g\_binlimit];

h1.Normalization = 'probability'; %Normalize data by total number of samples

g\_pmf = h1.Values;

title('Google Stock Data');

%title('Test Signal');

ylabel('Probability of Stock Values');

xlabel('Stock Values');

figure(2)

subplot(1,1,1);

hold on;

%Plot histogram and alter specifications

h2 = histogram(a\_sig,'BinWidth',a\_bin); %change widht of the bins

h2.BinLimits = [-32767 32767]; %Alter limits of historam

h2.Normalization = 'probability'; %Normalize data by total number of samples

a\_pmf = h2.Values;

title('Speech Signal');

ylabel('Probability of Amplitude Value');

xlabel('Signal Amplitude');

**The fisrt part of the code creates the histograms of the data sets the histograms are made using a bin size of 25 for the Google stock and 5 for the audio signal. These values were chosen from trial and error to determine which sizes would allow for all bins to be filled.**

%%

%Part 2

%Initialize vecctors for mean and variance

g\_mean = zeros(1,length(g\_pmf));

g\_cmoment = zeros(1,length(g\_pmf));

a\_mean = zeros(1,length(a\_pmf));

a\_cmoment = zeros(1,length(a\_pmf));

%get bin edges in order to find middle values of bins

gx = h1.BinEdges;

ax = h2.BinEdges;

%initialize vectors to hold middle values of bins

gx2 = zeros(1,length(gx));

ax2 = zeros(1,length(ax));

%find middle values of bins and store them in a vector

for i = 1:(length(gx)-1)

 gx\_buffer = floor((gx(i+1)-gx(i))/2);

 gx2(i) = gx(i)+gx\_buffer;

end

for i = 1:(length(ax)-1)

 ax\_buffer = floor((ax(i+1)-ax(i))/2);

 ax2(i) = ax(i)+ax\_buffer;

end

%Fill vectors with appropiate values

for i = 1:length(g\_pmf)

 g\_mean(i) = (gx2(i))\*g\_pmf(i);

end

for i = 1:length(a\_pmf)

 a\_mean(i) = (ax2(i))\*a\_pmf(i);

end

%Calculate variance of Google Stock

g\_expect2 = mean(g\_stock); %Mean function to have reference value

g\_expect = sum(g\_mean);

for i = 1:length(g\_pmf)

 g\_cmoment(i) = ((gx2(i)-g\_expect).^2)\*g\_pmf(i);

end

g\_var2 = var(g\_stock); %Variance function to have reference value

g\_var = sum(g\_cmoment);

%Calculate variance of audio signal

a\_expect2 = mean(a\_sig); %Mean function to have reference value

a\_expect = sum(a\_mean);

for i = 1:length(a\_pmf)

 a\_cmoment(i) = ((ax2(i)-a\_expect).^2)\*a\_pmf(i);

end

a\_var2 = var(a\_sig); %Variance function to have reference value

a\_var = sum(a\_cmoment);

ge2 = mean(g\_pmf);

%Get Gaussian Distrobutions of data

g\_stdv = sqrt(g\_var);

a\_stdv = sqrt(a\_var);

google\_Gaussian = normpdf(gx2,g\_expect,g\_stdv);

audio\_Gaussian = normpdf(ax2,a\_expect,a\_stdv);

%Plot the distrobutions ontop of the histograms

figure(1)

subplot(1,1,1),plot(gx2,google\_Gaussian);

figure(2)

subplot(1,1,1),plot(ax2,audio\_Gaussian,'.');

pause;

**Part 2 of the code calculates the mean, variance, and fits a Gaussian distribution over each data set. The middle values of the bins are used for the calculations as well as for finding the Gaussian PDF of each data set.**

%%

%Part 3

k = 4;

g\_erlang\_buffer = zeros(length(k),length(gx2));

g\_erlang = zeros(1,length(gx2));

 for i = 1:length(k)

 g\_erlang = ((gx2.^(k-1).\*exp(-gx2./g\_expect))./((g\_expect.^k)\*(factorial(k-1))))';

 g\_erlang\_buffer(i,:) = g\_erlang;

 end

 sample\_mean = poissfit(gx2);

 g\_poiss = poisspdf(gx2,sample\_mean);

%close 1;

figure(1);

subplot(1,1,1),plot(gx2,g\_poiss,'g');

subplot(1,1,1),plot(gx2,g\_erlang,'c');

hold on;

legend('Histogram','Gaussian Distribution','Poisson Distribution','Erlang Distribution');

**Part 3 of the MATLAB code finds alternate distribution fits for the Google stock data an plots them overtop the histogram and Gaussian distribution. The Poisson and Erlang distributions are used as the alternate distributions.**

# Conclusions

The Gaussian distribution fit the Audio signal because it is a real world signal, and in the real world, “everything is a Gaussian.” The signal has the most probability concentrated in the middle bins which gives it a Gaussian shape. However, the Google Stock data does not take on this shape. The Gaussian distribution for this data set, while a poor fit, is still the best distribution for this data. This is because once again, “everything is a Gaussian.” This data set is also real world, however its values are even more random than the audio signal due to the way in which stock is calculated The Google Stock is more consistent in the early time values, giving it the most concentration in those x-values. However, because stock values change sporadically, there is no set trend or distribution that can accurately depict the distribution of these values. Because there are a large number and each stock value appears a small number of times, the probabilities of each bin when using each distribution are very low, resulting the in the graph shown in figure 3 where each distribution appears to be too small for the histogram.