[Haiming Xu]

ECE 3512: Signals – Continuous and Discrete

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19122

# Problem Statements

This assignment is to demonstrate how to model data using a parametric model of a pdf. We need to compute a histogram of the amplitude of the Google stock and speech data and normalize it by the number of samples so that it is an estimate of the pdf. Then fit the distribution by estimating the mean and variance. Plot the Gaussian model on top of the histogram. Compare and contrast the quality of the fits to the data. Select another distribution from Chapter 4 that provides a better estimate of both data. Plot this model on the same graph with the histogram and the Gaussian fit. Compute the mean-squared error between the actual data and the parametric fit.

# Approach and Results

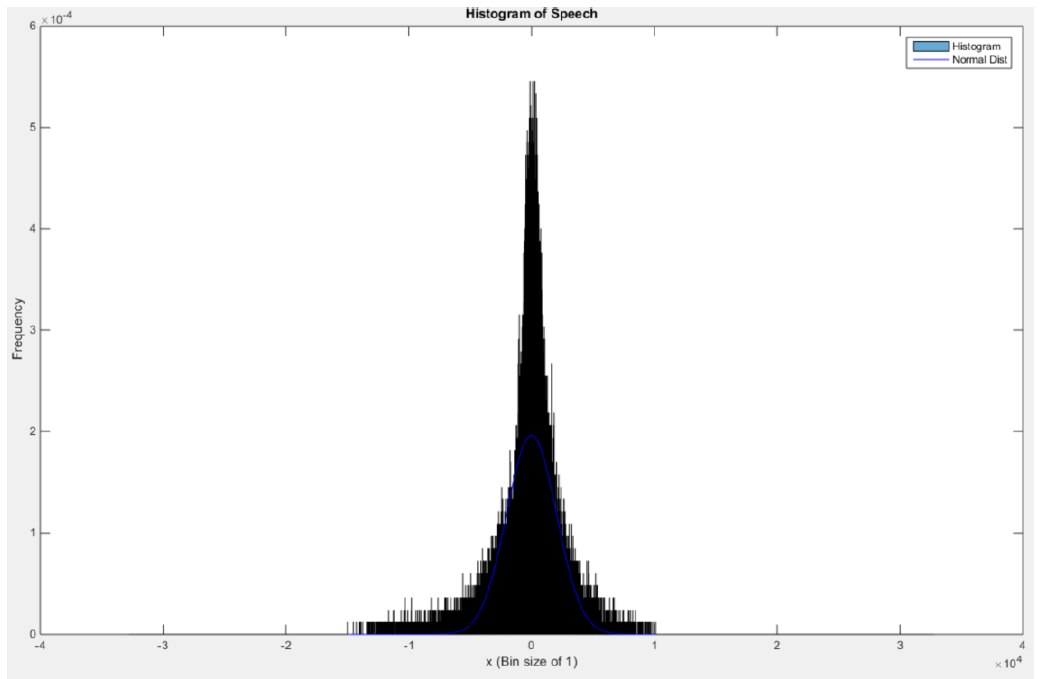


Figure 1: A histogram plot of the speech signal also showing a normal distribution fit.

First we compute the histogram plot of the speech signal. From the graph we can see it is a narrow bell sharp. When we fit the normal distribution in to the graph, we can see they match well. As we found in last computer assignment that most of the sample of speech signal stays around the zero as a sine wave and the amplitude depends on the voice change of the speech.

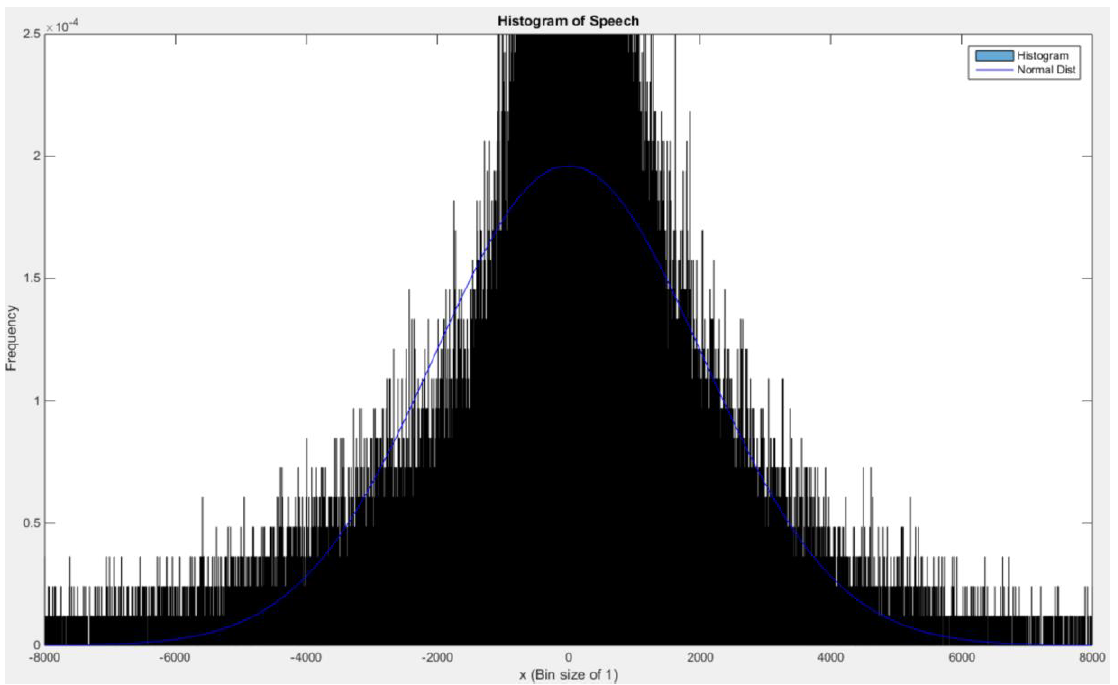
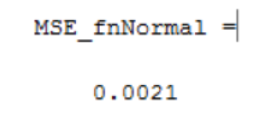


Figure 2: A histogram plot of the speech signal also showing a normal distribution fit (Zoomed in to sample value ranging: [-6000:6000])

In our analysis we compared the histogram to normal distribution fit to find a mean squared error between the two.



When we zoom in the graph from [-8000,8000], we can see most of our sample data distribution fits the normal distribution. As the means square error is extremely small as 0.0021, we can say that normal distribution can represent the audio speech signal. However, the distribution cannot predict the peak of the audio signal when the audio signal has a peak of 5.5\*10^-4 but the normal distribution only has a peak of 2 \*10^-4. At the same time distribution in the outer band smaller than -6000 and greater than 6000 cannot be predicted by the normal distribution either.

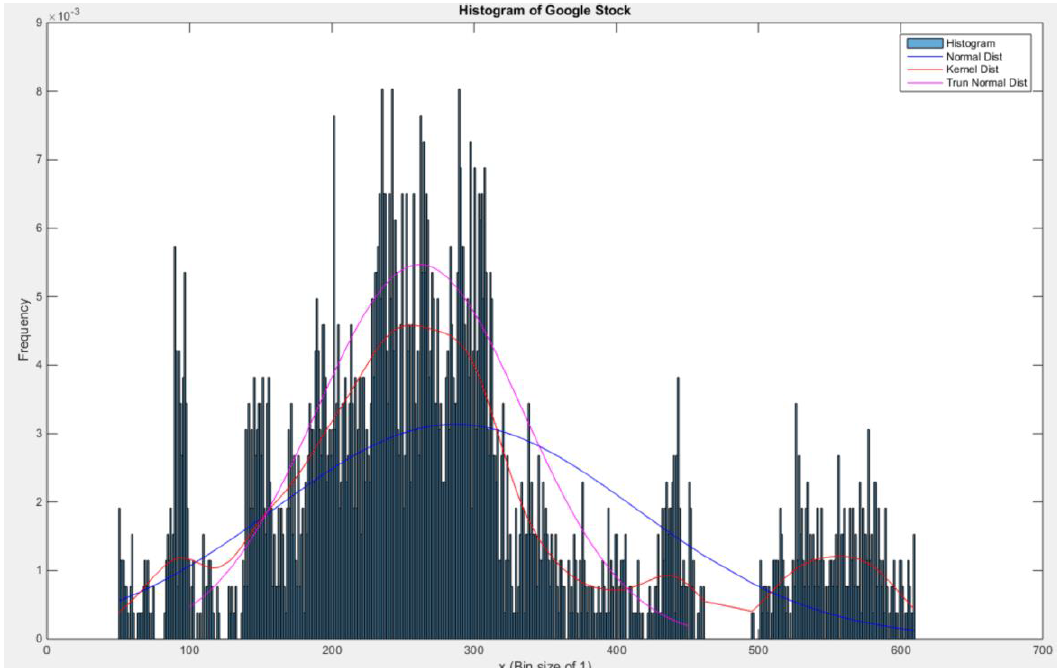
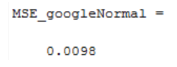


Figure 3: A histogram plot of the google stock also showing three distributions fits.

The second data is the google stock price, which is highly not match its normal distribution. From the last computer assignment we know that the mean of google stock price data set is 286 and the variance is 16194. The normal distribution graph we plot using these data is showed in blue line. As it has a big difference between the normal distribution and the google stock data. By computing the mean square error using the mean and the variance we got



Which is not really good error range too. Hence we tried some other distribution method. Firstly the truncated normal distribution, as is showed in the plot as a magenta line. It matched very well in the lower mean part of data, however, it did not take the data from range 500 to 600 under cover.

Also, we tried Kenel distribution using the data, which is showed as a red line in the plot. It highly represent most of the data distribution in that case.

# MATLAB Code

%%

clear; clc; close all;

% Let's first open the raw speech data file and store its values in a

% vector fn %

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

% Test Sine Wave

%fp=fopen('rec\_01\_sine.raw', 'r');

fn=fread(fp,inf,'int16');

fclose(fp);

L\_speech = length(fn);

% We are given a sample frequency of 8 kHz

fs = 8000;

L\_speech = length(fn);

timeL = L\_speech/fs;

% We can find the legnth of our signal given our sample frequency

t= linspace(0, timeL,L\_speech);

% Let's open the xls data file and store its values in avector fn

%

google\_v00 = xlsread('google\_v00.xlsx');

% google\_open = google\_v00(:,1);

% google\_high = google\_v00(:,2);

% google\_low = google\_v00(:,3);

google\_close = google\_v00(:,4);

L\_googleClose = length(google\_close);

clear google\_v00

% Let us find the min/max val, mean, median, and variance

google\_min = min(google\_close);

google\_max = max(google\_close);

google\_mean = mean(google\_close);

google\_median = median(google\_close);

google\_var = var(google\_close);

fn\_min = min(fn);

fn\_max = max(fn);

fn\_mean = mean(fn);

fn\_median = median(fn);

fn\_var = var(fn);

% Print our findings

out = sprintf('Google data: min = %f, max = %f, mean = %f, median = %f, variance = %f\n'...

, google\_min, google);

out = sprintf('Speech data: min = %f, max = %f, mean = %f, median = %f, variance = %f\n'...

, fn\_min, fn\_max, fn\_mean, fn\_median, fn\_var);

disp(out);

bin\_size = 1;

% ------------------ Speech ------------------

% Compute Histogram and plot histogram

bounds = [-32768 round(fn\_min/5)\*5:bin\_size:round(fn\_max/5)\*5 32768];

figure('name','[ECE 3522] Class Assignment [2]');

h\_fn = histogram(fn, bounds, 'Normalization', 'probability');

title('Histogram of Speech');

xlabel(sprintf('x (Bin size of %d)',bin\_size));

ylabel('Frequency');

hold on

% Sort the Speech Data

fn\_sort = sort(fn);

% Upsample the histogram frequencies

index = 1;

hfnUpSam = zeros(L\_speech, 1);

hist\_fnValue = h\_fn.Values;

for D = floor(fn\_min):bin\_size:floor(fn\_max)-1

for k = 1:L\_speech

if ((D <= fn(k)) && (fn(k) < D+1))

hfnUpSam(k) = hist\_fnValue(index);

end

end

index = index + 1;

end

% Use a function to fit the data to a Normal distribution

pd = fitdist(fn\_sort, 'Normal');

% Find values of the PDF for our data

yfnNormal = pdf(pd,fn\_sort);

% Plot the Normal Distribution

plot(fn\_sort, yfnNormal, 'b');

% Label the axis

legend('Histogram', 'Normal Dist');

% Find MSE for the Normal on Speech Signal

for k = 1:L\_speech

xt(k) = (yfnNormal(k)-hfnUpSam(k))^2;

end

MSE\_fnNormal = sum(xt);

hold off

% ------------------ Google ------------------

clear xt pd

% Compute Histogram and plot histogram

bounds = [round(google\_min/5)\*5:bin\_size:round(google\_max/5)\*5];

figure('name','[ECE 3522] Class Assignment [2]');

h\_google = histogram(google\_close, bounds, 'Normalization', 'probability');

title('Histogram of Google Stock');

xlabel(sprintf('x (Bin size of %d)',bin\_size));

ylabel('Frequency');

hold on

% Upsample the histogram frequencies

index = 1;

hGoogleUpSam = zeros(L\_googleClose, 1);

hist\_googleValue = h\_google.Values;

for D = floor(google\_min):bin\_size:floor(google\_max)-1

for k = 1:L\_googleClose

if (D <= google\_close(k) && google\_close(k) < D+1)

hGoogleUpSam(k) = hist\_googleValue(index);

end

end

index = index + 1;

end

% Google Normal Distribution Computation

% Sort the Google Data

google\_close\_sort = sort(google\_close);

% Use a function to fit the data to a Normal distribution

pd = fitdist(google\_close\_sort, 'Normal');

% Find values of the PDF for our data

yGoogleNormal = pdf(pd,google\_close\_sort);

% Plot the Normal Distribution

plot(google\_close\_sort, yGoogleNormal, 'b');

% Find MSE for the Normal on GoogleClose

for k = 1:L\_googleClose

xt(k) = (yGoogleNormal(k)-hGoogleUpSam(k))^2;

end

MSE\_googleNormal = sum(xt);

% Find the truncated Normal Distribution

for i = 1:length(google\_close\_sort)

if (google\_close\_sort(i)) <= 100

indexMin = i;

end

if (google\_close\_sort(i)) >= 450

indexMax = i;

break

end

end

% Find the new variance and Means for the truncated GoogleClose

google\_trun\_var = var(google\_close\_sort(indexMin:indexMax));

google\_trun\_mean = mean(google\_close\_sort(indexMin:indexMax));

% Use a function to fit the data to a Normal distribution

pd = makedist('Normal', 'mu',google\_trun\_mean,'sigma',sqrt(google\_trun\_var));

yGoogleTrunNormal = pdf(pd,google\_close\_sort(indexMin:indexMax));

plot(google\_close\_sort(indexMin:indexMax), yGoogleTrunNormal, 'm');

legend('Histogram', 'Normal Dist', 'Kernel Dist', 'Trun Normal Dist');

# Conclusions

In this assignment we tried some different distribution model to analysis the speech audio signal and the Google stock price data set. And notice that an actual application is when finding the rate of defection for products coming off an assembly line. We can feed the data into a distribution model and predict the rate of failing products. Industry also use this method to allocating a budget for servicing these defective products.