Miles A. Vendetti-Houser

ECE 3522: Stochastics

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

# Problem Statement

The assignment is designed to demonstrate how data can be modeled using parametric models of a probability density function. In statistics, a parametric model is a family of distributions that can be described using a finite number of parameters.

# Approach and Results

The first task accomplished required computing a histogram. The amplitude of the data needed to be normalized by the total number of samples. In order to complete this task, I used my resources wisely as throughout the computer assignments, I am realizing that MATLAB provides a ton of useful built-in functions. May as well take advantage of them! After browsing through the available inventory, I stumbled upon a function named histcount().

This function supports ‘name-value’ pair options. One of the name-value pair options allowed the normalization to be defined. I changed the normalization parameter to ‘pdf’ where MATLAB doc states this is the “*Probability density function estimate. Each N value is equal to the relative number of observations in the bin divided by the bin width*”. The two figures shown below display the resulting bar graph plots of the histogram data. The figure on the left shows the histogram for the audio signal. The figure on the right is the Google closing data.





For problem two in the computer assignment, we had to fit a normal (Gaussian) distribution to the histogram plots from part 1. I took advantage of some built-in functions once again in which resulted with the two figures shown below. More explanation to come!



The two figures above contain the normal distributions along with a second distribution fit. For part three, it was necessary to select another distribution that better fits the histograms. The Google PDF needed something other than the Gaussian dist. Visually speaking, it did not look appropriate. We need to find a curve that better describes the full distribution of possible closing values for the stock prices. After some necessary research, I came across the family of Gamma distributions. Where we see probability distributions covered in the text such as the exponential, Poisson, chi-squared, and my bread winner Erlang distribution. I chose this curve fit because the gamma distribution is referred to as an Erlang distribution when the coefficients are all positive integers. As we can tell, the blue curve (Erlang) better represents the histogram than the red normal curve on the Google figure above.

As for the audio signal, I went with a Student-T Distribution to better define the data. After ready about this fit on the interweb, I found out that the student-t holds its usefulness when observing data with additive errors. In other words, if the standard deviation of the errors were known, the normal distribution would be a better fit for the data.

However, if we were taking real-time data or in some setting where the data observed is exposed to errors, the standard deviation of the errors would be unknown. The t-distribution accounts for this uncertainty, which is why I think it qualifies as a more appropriate fit for the audio signal histogram shown above.

# MATLAB Code

% Computer Assignment 04

%

clear;clc;clf;close all;

%%%%%%% Import Block %%%%%%%%%

filename = 'rec\_01\_speech.raw';

ff1 = fopen(filename);

sig = fread(ff1, inf, 'int16');

fclose(ff1);

filename = 'google\_v00.xlsx';

[~,~,raw] = xlsread(filename);

if(ispc)

 formatIn = 'mm/dd/yyyy';

 data\_date = datenum(char(raw(2:end,1)), formatIn);

elseif(isunix)

 data\_date = cell2mat(raw(2:end,1));

end

data\_close = cell2mat(raw(2:end,5));

%%%%%%% Audio Signal %%%%%%%%%

%%%%% PART ONE

binwidth1 = 10;

norm1 = 'pdf';

numBins1 = round(length(sig) / binwidth1);

binlimits1 = [-17767, 17767];

[N1, edges1, bins1] = histcounts(sig, numBins1, 'BinLimits', binlimits1, ...

 'Normalization', norm1, 'BinWidth', binwidth1, 'BinMethod' , ...

 'integers');

figure(1)

bar(edges1(:,1:end-1),N1)

axis([binlimits1(1) binlimits1(2) 0 max(N1)])

xlabel('Signal Amplitudes')

ylabel('Frequency')

title('Histogram')

%%%%%%% Google Stock %%%%%%%%%

%%%%% PART ONE

% binwidth = 10;

% norm = 'pdf';

% numBins = round(length(data\_close) / binwidth);

% binlimits = [0 700];

% [N2, edges2, bins2] = histcounts(data\_close, numBins, 'BinLimits', binlimits, ...

% 'Normalization', norm, 'BinWidth', binwidth, 'BinMethod' , ...

% 'integers');

% figure(2)

% hold on

% bar(edges2(:,1:end-1),N2)

%

% hold off

% axis([binlimits(1) binlimits(2) 0 max(N2)])

% xlabel('Closing Prices')

% ylabel('Frequency')

%

lambda = mean(data\_close)

data\_close = data\_close(:);

% Code Generated from DISTTOOL Function

% Prepare figure

clf;

hold on;

LegHandles = []; LegText = {};

% --- Plot data originally in dataset "Google PDF"

[CdfF,CdfX] = ecdf(data\_close,'Function','cdf'); % compute empirical cdf

BinInfo.rule = 5;

BinInfo.width = 10;

BinInfo.placementRule = 2;

BinInfo.anchor = 700;

[~,BinEdge] = internal.stats.histbins(data\_close,[],[],BinInfo,CdfF,CdfX);

[BinHeight,BinCenter] = ecdfhist(CdfF,CdfX,'edges',BinEdge);

hLine = bar(BinCenter,BinHeight,'hist');

set(hLine,'FaceColor','none','EdgeColor',[0.333333 0.666667 0],...

 'LineStyle','-', 'LineWidth',1);

xlabel('Data');

ylabel('Density')

LegHandles(end+1) = hLine;

LegText{end+1} = 'Google PDF';

% Create grid where function will be computed

XLim = get(gca,'XLim');

XLim = XLim + [-1 1] \* 0.01 \* diff(XLim);

XGrid = linspace(XLim(1),XLim(2),100);

% --- Create fit "Erlang (Gamma)"

pd1 = fitdist(data\_close, 'gamma');

YPlot = pdf(pd1,XGrid);

hLine = plot(XGrid,YPlot,'Color',[0 0 1],...

 'LineStyle','-', 'LineWidth',2,...

 'Marker','none', 'MarkerSize',6);

LegHandles(end+1) = hLine;

LegText{end+1} = 'Erlang (Gamma)';

% --- Create fit "Gaussian (Normal)"

pd2 = fitdist(data\_close, 'normal');

YPlot = pdf(pd2,XGrid);

hLine = plot(XGrid,YPlot,'Color',[1 0 0],...

 'LineStyle','-', 'LineWidth',2,...

 'Marker','none', 'MarkerSize',6);

LegHandles(end+1) = hLine;

LegText{end+1} = 'Gaussian (Normal)';

% Adjust figure

box on;

hold off;

axis([0 700 0 (6\*10^-3)])

title('Google Stock')

% Create legend from accumulated handles and labels

hLegend = legend(LegHandles,LegText,'Orientation', 'vertical', ...

 'FontSize', 9, 'Location', 'northeast');

set(hLegend,'Interpreter','none');

% --- Speech Signal

% Prepare figure

figure(2)

hold on;

sig = sig(:);

LegHandles = []; LegText = {};

% --- Plot data originally in dataset "Speech Signal"

[CdfF,CdfX] = ecdf(sig,'Function','cdf'); % compute empirical cdf

BinInfo.rule = 4;

BinInfo.width = 10;

[~,BinEdge] = internal.stats.histbins(sig,[],[],BinInfo,CdfF,CdfX);

[BinHeight,BinCenter] = ecdfhist(CdfF,CdfX,'edges',BinEdge);

hLine = bar(BinCenter,BinHeight,'hist');

set(hLine,'FaceColor','none','EdgeColor',[0 .1 1],...

 'LineStyle','-', 'LineWidth',1);

xlabel('Data');

ylabel('Density')

LegHandles(end+1) = hLine;

LegText{end+1} = 'Speech Signal';

% Create grid where function will be computed

XLim = get(gca,'XLim');

XLim = XLim + [-1 1] \* 0.01 \* diff(XLim);

XGrid = linspace(XLim(1),XLim(2),100);

% --- Create fit "Normal"

% Fit this distribution to get parameter values

pd3 = fitdist(sig, 'normal');

YPlot = pdf(pd3,XGrid);

hLine = plot(XGrid,YPlot,'Color',[ 0.5 1 .1],...

 'LineStyle','-', 'LineWidth',2,...

 'Marker','none', 'MarkerSize',6);

LegHandles(end+1) = hLine;

LegText{end+1} = 'Normal';

% --- Create fit "Students t"

% Fit this distribution to get parameter values

pd4 = fitdist(sig, 'tlocationscale');

YPlot = pdf(pd4,XGrid);

hLine = plot(XGrid,YPlot,'Color',[1 0 1],...

 'LineStyle','-', 'LineWidth',2,...

 'Marker','none', 'MarkerSize',6);

LegHandles(end+1) = hLine;

LegText{end+1} = 'Students t-distribution';

% Adjust figure

box on;

hold off;

title('Speech Signal')

axis([-5000 5000 0 (4\*10^-4)])

% Create legend from accumulated handles and labels

hLegend = legend(LegHandles,LegText,'Orientation', 'vertical', ...

 'FontSize', 9, 'Location', 'northeast');

set(hLegend,'Interpreter','none');

# Conclusions

After completing this computer assignment, I have gained a better understanding of why we need to decide on the most appropriate curve fit. The probability distributions assign a probability value to all of the measureable subsets of possibly outcomes for random experiments. The distributions each look at different variables of the data whether it’s the variance, the root-mean, the frequency, the number of positive occurrences, etc.