Anton LeKang

ECE 3522: Stochastic Processes in Signals and Systems

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

CA4

# Problem Statement

The objective of this lab is to normalize a histogram graph, and then plot the normal distribution or Gaussian distribution. The Gaussian distribution tells the probability that data value will fall between two points. This distribution is not always the best one to use, especially if there are multiple peeks in the plot. Another objective of this lab is to find a distribution that that is the best fit for the data. Then the mean squared error can be computed from the different distributions. This computer assignment will use Google’s stock prices, and an audio file as the main inputted source of the data. MATLAB will be used to compute each distribution, as well as the mean squared error.

# Approach and Results

Part 1 and 2: Plot the histogram of the data, plot the Gaussian model on top of the histogram

The first part of the computer assignment shows the plot of the histograms for each of the data samples.

The second part of the computer assignment plots the Gaussian distribution over the normalized histogram. This curve is shown by the red in figure 1, and figure 2. The Gaussian line is seen better in the Google stock prices, but it does not fit perfectly with the curve. In figure 2, the Gaussian line did a shade of the normal distribution. The audio file seemed to be acting up when I kept trying to plot the Gaussian. Even though it is really small, if it were scaled properly it would fit the data output nicely.

Figure 1 – Normalized Google Stock with Gaussian

Figure 2 – Normalized Audio file with Gaussian

Part 3: Finding a plot that fits the distribution and MSE

Part 3 of the computer assignment is asked to find a better distribution for the each of the data files. The book did not have that many good choices, because they all had specific guidelines for the data. I decided to use a Kernel distribution for the Googles stock price and a fitted distribution for the Audio file.

Figure 3 – Kernel Distribution with Google Stock

Figure 4 – Fit normalized distribution with audio file

When looking at the figures, both distributions fit better to the data. In figure 3, the Kernel distribution fits very nicely with the output of the data, and goes up where there are peaks. Figure 4 shows the normalized fit distribution which should have happened in part 1 and 2 of this computer assignment. The distribution fits nicely with the output of the data. Next the mean squared error was computed of each plot. The mean squared error looks at the mean of the distribution and compares it to the mean of the signal. This is very useful when trying to decide which distribution to use. The mean square error of the Google stock was extremely low at 2.3284e-10. This is surprising to me because that is almost 0%, and should not happen. Either the distribution fits the mean 100%, or something is messed up with the distribution. The mean squared error of the Audio file was, 0.1514. This is a more accurate mean squared error, and shows that the distribution is fitting nicely with the curve.

# MATLAB Code

Google Stock Prices

function CA4\_part2

clear all

clc

[sig, txt, raw] = xlsread('google\_v00.xlsx', 1);

sig = sig(:,4);

% load the data:

% It is 16-bit sampled data, so we must load it as short integers.

%

[z,q] = size(sig)

mn = mean(sig)

vr = nanvar(sig)

sqr = sqrt(vr)

norm = normpdf(sig, mn, sqr);

xbins = (-32767:1:32767);

x\_values = -32767:1:32767;

pd = fitdist(sig,'Kernel','Kernel','epanechnikov')

y = pdf(pd,x\_values);

mn1 = mean(y)

figure(1)

h = hist(sig, xbins)

h = h/z

stem(xbins, h, 'b')

xlim([0, 650]);

hold on

plot (sig, norm, 'r')

hold on

plot(x\_values, y, 'k', 'linewidth', 3)

xlabel('Normalized Frequency')

ylabel('Index')

title('Google Stock Gaussian and Kernal Distribution')

legend('Histogram', 'Gaussian', 'Kernal Distribution', 'location', 'northwest')

perf = mse(mn1,y, sig)

end

Audio File:

function CA4\_part4sig

clear all

clc

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

% load the data:

% It is 16-bit sampled data, so we must load it as short integers.

%

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

[z,q] = size(sig)

mn = mean(sig)

vr = nanvar(sig)

sqr = sqrt(vr)

norm = normpdf(sig, mn, sqr);

xbins = (-32767:10:32767);

x\_values = -32767:10:32767

pd = fitdist(sig,'Normal') %fitting normal dist

y = pdf(pd,x\_values);

mn1 = mean(y)

% figure(1)

% h = hist(sig, xbins)

% y = h/z

%

% stem(xbins, y)

%

% hold on

% plot(sig, norm, 'r')

% hold on

% plot(x\_values,y,'y', 'LineWidth',2)

% xlabel('Normalized Frequency')

% ylabel('Index')

% legend('Histogram', 'Gaussian', 'Normalized Fit', 'location', 'northwest')

% title('Audio Signal Gaussian and Normalized Fit')

% %end

perf = mse(mn,y,sig)

end

Both of these codes were relatively the same, except for the bin sizes, and the extra distribution graph. The bin sizes for the Google prices were smaller, and the total number of values was also smaller. For the Google stack prices the Fitdist(‘Kernal’) instruction is how the Kernel graph was made, and then the pdf of that signal was taken. Similarly the normal value of the audio file was taken. The mean squared error was computed at the end of each of the scripts, by using the means of the signal, with the means of the distribution.

# Conclusions

Distributions are a very valuable tool when looking at and predicting future outputs of data. The Gaussian is the king of all distributions and is very useful in a lot of statistical analyses. It is so profound because it looks at the mean and the variance, and comes up with a normalized curve. The Gaussian looks really good in figure 4, which is the audio file. The Gaussian is ok for the Google stock prices, but does not fit it exactly. The better fit for the Google stock price was the Kernel Distribution. The Kernel Distribution basically sets the pdf of the graph at hand. After plotting the distributions the mean squared error was computed. Each of the distributions got very good results and low error values. This is good because the mean square looks at the error between the distribution and mean. This computer was very useful in learning about distribution and how they relate to signals. It shows how valuable some distributions are compared to other ones, based on the output. This is very useful in statistical analysis.