TEMPLE UNIVERSITY COLLEGE OF ENGINEERING | ECE 3623 | SPRING 2015

Computer Assignment (CA) No. 4: Model Fitting

Tyler Berezowsky

February 22, 2015

1 Problem Statement

This assignment introduced modeling real world distributions in MATLAB through the normal distribution, and expands to representing the data with other distributions. This was accomplished through the three tasks below:

1. Compute a histogram of the amplitude of the data and normalize it by the number of samples so that it is an estimate of the pdf.
2. Fit this 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.
3. In (2), you should find that the Gaussian model is not a good fit for the Google data. Select another distribution from Chapter 4 that provides a better estimate of the 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. Which gives a better approximation? (Do this for both data sets.)

2 Approach and Results

Task 1

A histogram for each data set was generated through the MATLAB command `histogram` with arguments to automatically normalize the distribution. This could also be accomplished by dividing the amplitude of each bin via the total number of bins in the histogram.

Task 2

The mean and variance of each data set was calculated. The parameters were then substituted into the normal distribution, equation 2.1. The normal distribution was then graphed over of the histogram of each data set.

$$f(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.1}$$

Task 3

In the search for a better fitting distribution, each data set was loaded into the MATLAB's `dfittool`. The `dfittool` allows the user to quickly produce a variety of distributions to a data set. Through this tool the lognormal distribution was selected for Google data set. The lognormal distribution, also known as the Galton distribution after Francis Galton, is display in equation 2.2. [1]

$$f(x,\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{(\ln x-\mu)^2}{2\sigma^2}} \tag{2.2}$$

The distribution was then plotted over the histogram and normal distribution for the Google data set. The t Location-Scale distribution was selected as the distribution for the audio signal. The distribution is described below in equation 2.3 where υ is shape parameter and greater than 0.[2] The distribution was plotted over the histogram and the normal distribution of for the data set. The plots of the histogram, the normal distribution and the alternative distribution can be seen for the audio signal and the stock price in figures 2.1 and 2.2 respectively.

$$f(x,\mu,\sigma,\upsilon) = \frac{\Gamma\left(\frac{\upsilon+1}{2}\right)}{\sigma\sqrt{\upsilon\pi}\,\Gamma(\upsilon/2)} \left[\frac{\upsilon+\left(\frac{x-\mu}{\sigma}\right)^2}{\upsilon}\right]^{-\left(\frac{\upsilon+1}{2}\right)} \tag{2.3}$$

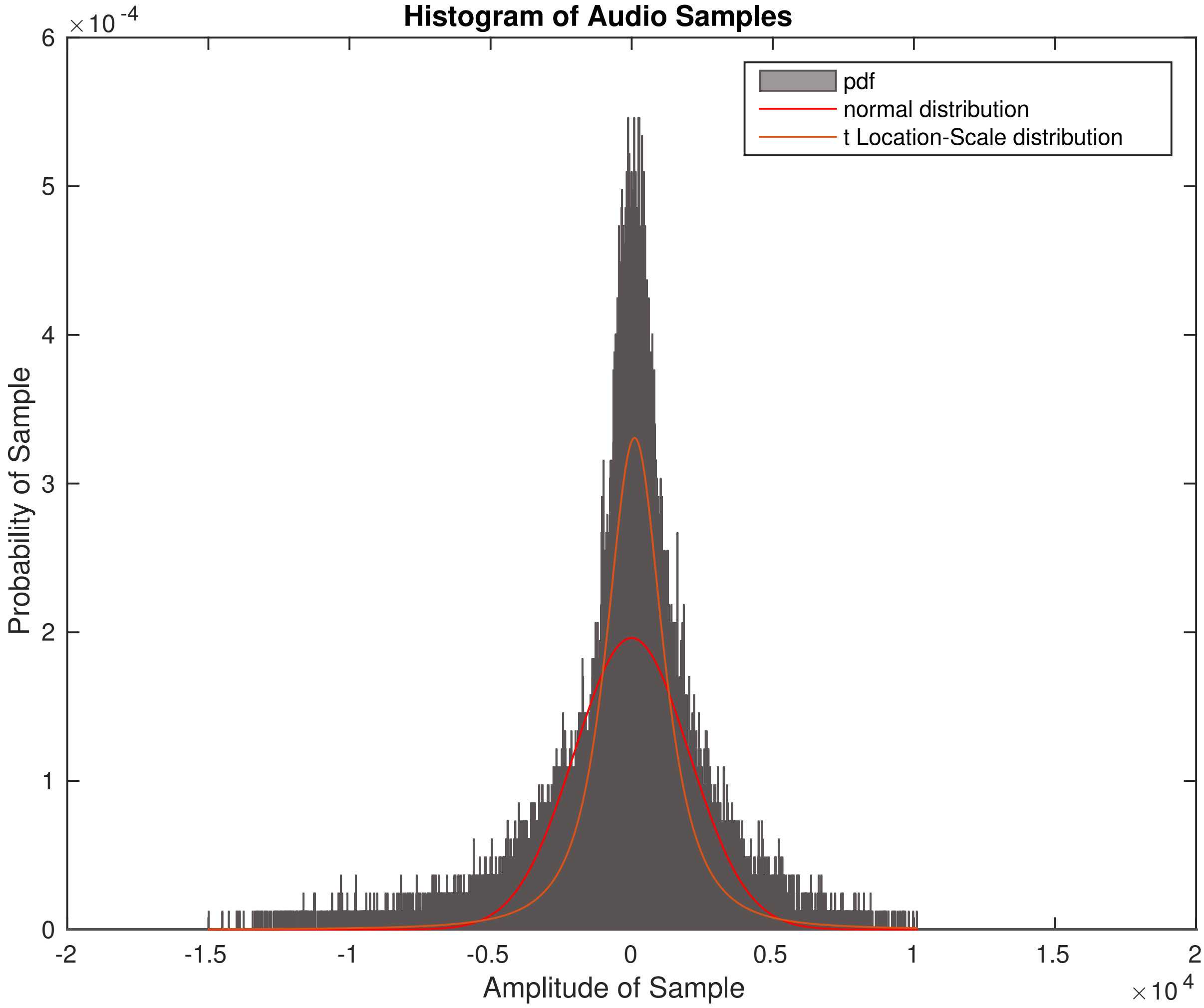

Figure 2.1: Histogram plot of audio signal (grey) with normal distribution (red) and t-Location-Scale distribution (orange).

[1] http://en.wikipedia.org/wiki/Log-normal_distribution

[2] http://www.mathworks.com/help/stats/t-location-scale-distribution.html?refresh=true

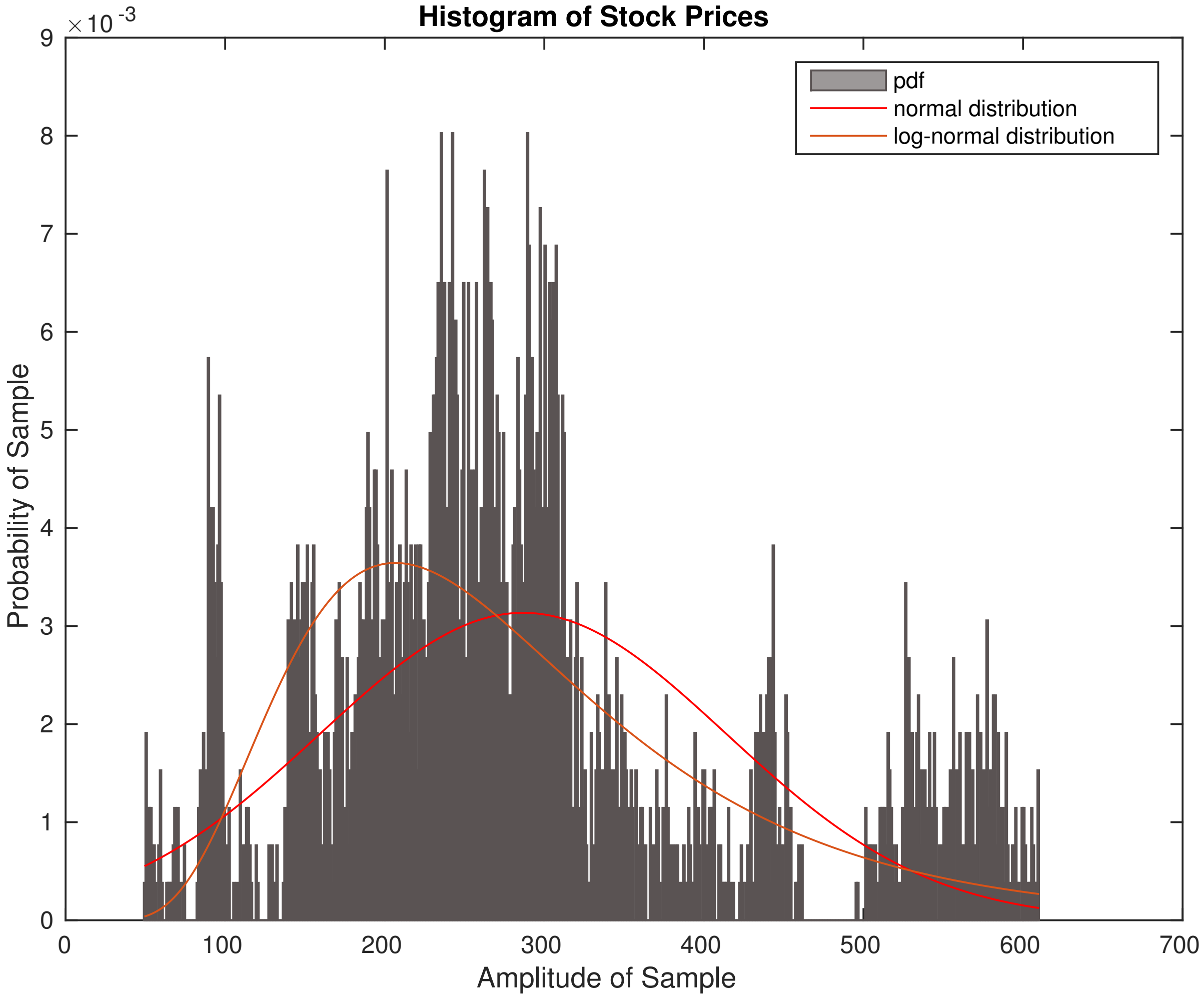

Figure 2.2: Histogram plot of google signal (grey) with normal distribution (red) and log-normal distribution (orange).

The mean squared error (MSE) was calculated between the histogram and the two distributions for each data set. The MSE error is described below where PMF_X is the histogram and f_X a parametrized distribution. The results are displayed in the table below.

$$MSE = E\left[(PMF_X - f_X)^2\right] \tag{2.4}$$

Audio Signal		Stock Price	
Normal:	5.82E-09	Normal:	2.16E-08
t Location-Scale:	9.43E-09	Lognormal:	1.94E-09

Table 2.1: MSE values for distributions trialed against each histogram.

3 MATLAB CODE

Listing 1: 'MATLAB solution for CA: 04.'

```
%% Computer Assignment 04
% 1) Compute a histogram of the amplitude of the data and normalize it by
% the number of samples so that it is an estimate of the pdf.
%
% 2) Fit this 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.
%
% 3) In (2), you should find that the Gaussian model is not a good
% fit for the Google data. Select another distribution from Chapter 4 that
% provides a better estimate of the 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. Which gives a better
% approximation? (Do this for both data sets.)

%% Import the Audio and Google datasets
% Import the excel spreadsheet into an array

clear; clc; clear all;

filename = 'google_v00.xlsx'
[data, header, raw] = xlsread(filename);

high = data(:,3);
low = data(:,4);
close = data(:,5);
open = data(:,2);
dates = x2mdate(data(:,1), 1);

% Import the .raw file into an array

filename = 'rec_01_speech.raw';
file = fopen(filename, 'r');
audio = fread(file, inf,'short');

%audio = audio(1:10000); % cut down audio!

sampleRate = 8e3
time = (0:length(audio)-1)*1/(sampleRate);


%% Nomarilze Data
% double check normalization
figure()
subplot(211)
plot(audio)
title('Audio Signal')
xlabel('Time (seconds)')
ylabel('Amplitude')
subplot(212)
plot(dates, close)
datetick('x',2 ,'keeplimits', 'keepticks');
title('Google Stock Price')
ylabel('Price')
xlabel('Date')

print -depsc2 time_plot.eps


%% Calculate Histograms
% Audio Signal
BINWIDTH = 1;
```

```matlab
fplota = figure()
h = histogram(audio, 'BinWidth', BINWIDTH, 'Normalization', 'probability', ...
    'BinLimits', [-32767 32767]);
h.EdgeColor = [0.3510, 0.3245, 0.3245];
h.FaceColor = [0.3510, 0.3245, 0.3245];
hista = h.Values;
ylabel('Probability of Sample');
xlabel('Amplitude of Sample');
title('Histogram of Audio Samples');
xlim([-2e4 2e4])

% Stock Price
BINWIDTH = 1;

fplotg = figure()
h = histogram(close, 'BinWidth', BINWIDTH, 'Normalization', 'probability');
h.EdgeColor = [0.3510, 0.3245, 0.3245];
h.FaceColor = [0.3510, 0.3245, 0.3245];
histg = h.Values;
ylabel('Probability of Sample');
xlabel('Amplitude of Sample');
title('Histogram of Stock Prices');

%% Calculate Mean and Variance for Modeling

% Audio Signal
meanA = mean(audio);
varA = var(audio);

% Stock Price
meanG = mean(close);
varG = var(close);

%% Generate and Plot Gaussian Model

% Audio Signal
a = linspace(min(audio), max(audio), length(hista));
f_a = 1./(sqrt(2*pi.*varA))*exp(-((a-meanA).^2)/(2*varA));

figure(fplota)
hold on;
plot(a, f_a, 'r')
legend('pdf', 'normal distribution')
print -depsc2 audio_plot.eps

% Stock Price
p = linspace(min(close), max(close), length(histg));
f_p = 1./(sqrt(2*pi.*varG))*exp(-((p-meanG).^2)/(2*varG));

figure(fplotg)
hold on;
plot(p, f_p, 'r')
legend('pdf', 'normal distribution')

%% Calculate Mean Squared Error Between Models and PDFs

msea_normal = mean((hista - f_a).^2)
mseg_normal = mean((histg - f_p).^2)

%% Overlay New Distribution Model for Google and Audio Data
% The stock price's distribution will not fit any "off the shelf"
% parameter distributions.
figure(fplotg)
```

```
pdG = fitdist(close,  'lognormal')
plot(p, pdG.pdf(p))
legend('pdf', 'normal distribution', 'log-normal distribution')
mseg_lognormal = mean((histg - pdG.pdf(p)).^2);
print -depsc2 google_plot.eps

figure(fplota)
pdA = fitdist(audio, 'tLocationScale')
plot(a, pdA.pdf(a))
legend('pdf', 'normal distribution', 't Location-Scale distribution')
msea_tLocationScale = mean((hista-pdA.pdf(a)).^2)
print -depsc2 audio_plot.eps
```

4 Conclusions

While there exists a collection of parametrized distributions, real data rarely conforms to the mold. This is reflected in both data sets. The histogram of the Google data set is particularly sporadic as it is a time-series with an increasing diversity for each sample. The audio signal is also a time-series, but with a constant predetermined range of samples.

Reducing the MSE with the selected distributions was surprisingly troubling. For the audio signal, the t Location-Scale was actually considerably worst increasing the MSE by 61%. This was surprising, as I initially thought the distribution's higher peak around the mean would reduce error substantially despite the distribution's lack of width compared to the PMF or normal distribution. If anything, this proves the need for criteria such as the MSE. The log-normal distribution reduced the MSE compared to the normal distribution by 10%.