

Eighth Assignment for Computational Physics

Xinyu Liu

November 5, 2024

Contents

1	My Github Page URL	2
2	Logistic regression	2
3	Newman 7.3: Fourier transformation for musical instrument	3
4	Newman 7.4: Fourier filtering and smoothing	7

1 My Github Page URL

<https://github.com/rising1227/phys-ga2000>

2 Logistic regression

In this problem we use a logistic model to fit the data, between ages and whether they know a certain phrase. In the code we first define our logistic model as a function. Then a loss function is defined as:

$$L = -p * \log(0.01 + 0.99 * q) - (1 - p) * \log(1 - 0.99 * q) + 0.0001 * (\beta_0^2 + \beta_1^2) \quad (1)$$

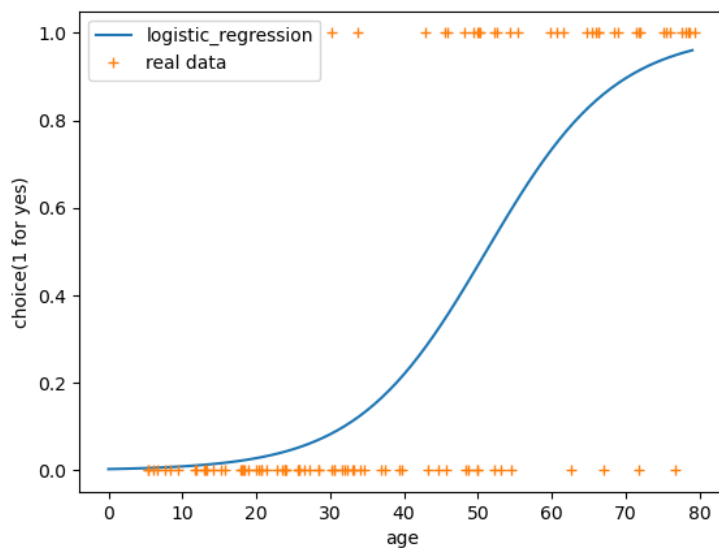
Where p is the real data and q is the predicted probability given by logistic model. We've modified the common negative log likelihood function by a little bit to avoid doing log for a value very close to 0. We also added a regularization term to avoid very large parameters.

After defining a loss function, we use a optimizor(`optimize.minimize`) to minimize this loss function and find the best fit parameter. In this process we use `jax diff` function to do automatic differentiation. We found that:

- The maximum likelihood values is: $\beta_0 = -5.83248073$, $\beta_1 = 0.11416234$
- The error for β_0 and β_1 is: `[1.162193 0.02321067]`
- The covariance matrix for β_0 and β_1 is: `[[1.3506925e+00, -2.5923811e-02], [-2.5923807e-02, 5.3873518e-04]]`

We can plot the real data and the logistic regression result in a same plot, here's the result.

Table 1: Real data and the logistic regression of the result



The result make sense to me. It shows that this phrase is more common in aged people. People with higher age has higher probability to know this phrase.

3 Newman 7.3: Fourier transformation for musical instrument

We use numpy to read the txt file into numpy array. This is plotted as follows:

Table 2: Waveform for piano

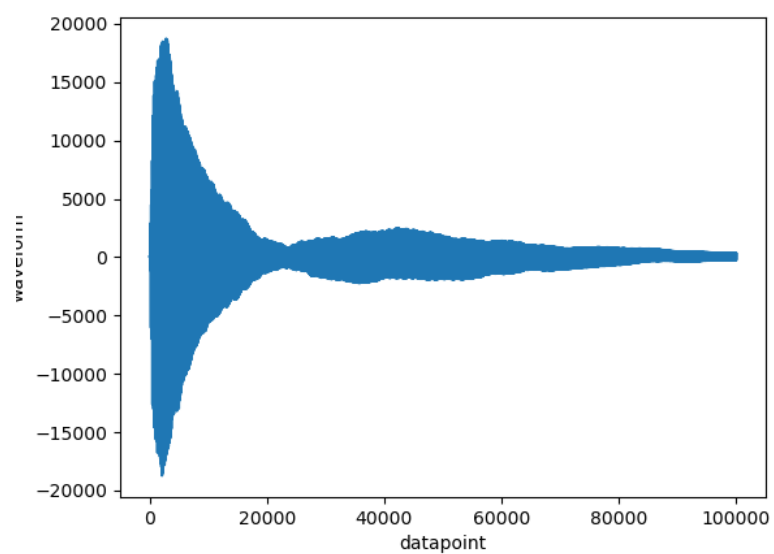
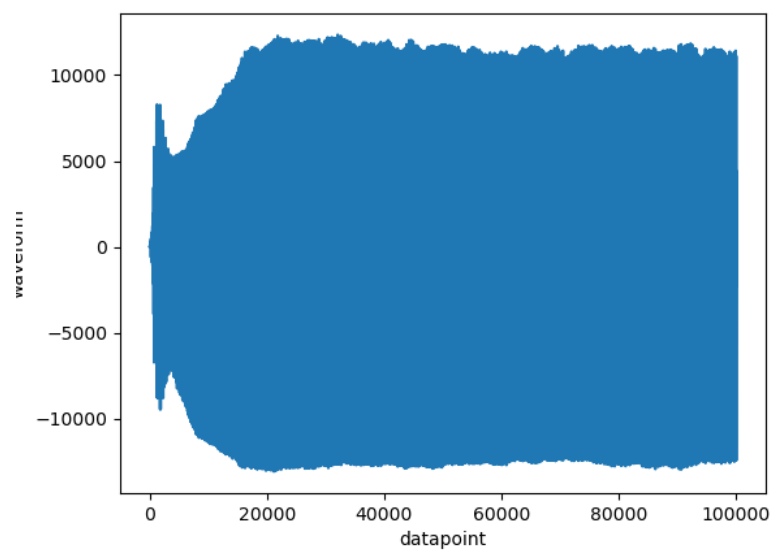


Table 3: Waveform for trumpet



In order to see some detail of the waveform, we plotted the number 5000-7000 data in the array (there's a total of 10^5 data point)

Table 4: Zoomed waveform for piano

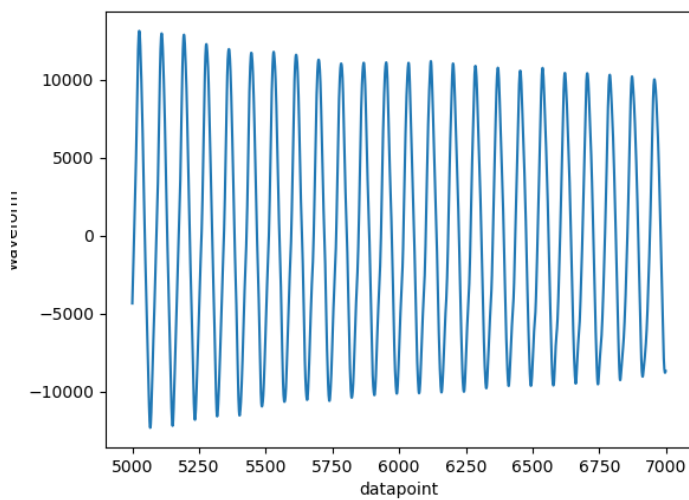
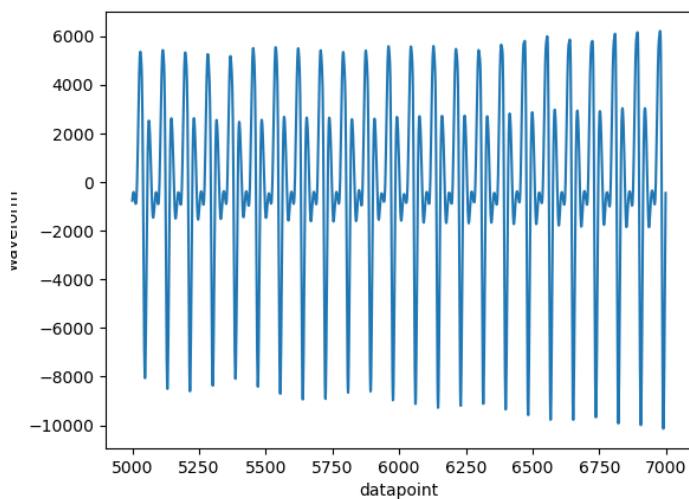


Table 5: Zoomed waveform for trumpet



We then do the FFT for the waveform.

Table 6: Fourier coefficient for piano waveform

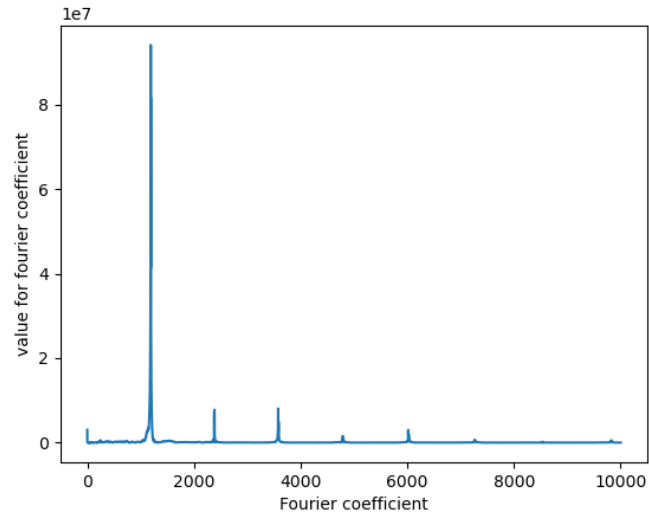
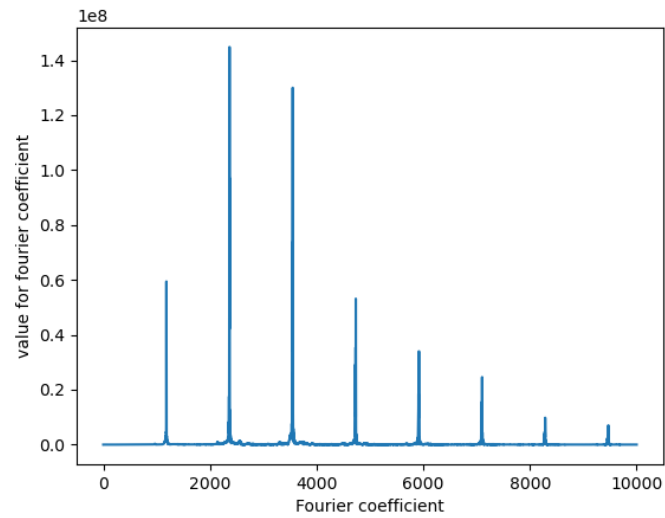


Table 7: Fourier coefficient for trumpet waveform



From the fourier coefficient, we can see that the waveform of a musical instrument has pretty pure frequency components and low noise. The frequency components are all integer number of a fixed frequency. This can be explained by the nature of stringed instrument, where the resonent frequency of the string is integer number of minimum frequency.

From the fourier coefficient of piano waveform and trumpet waveform, we can find the first peak value. For piano, we found that $n_1 = 1190$ and for trumpet $n_2 = 1184$. We know that the n^{th} fourier coefficient correspond to the waveform:

$$\phi(i) = e^{i*2\pi*n_i/N} \quad (2)$$

Now we know that the data aquisition speed is $f = 44100Hz$. This means that now for the n^{th} fourier coefficient, the waveform against time is:

$$\phi(t) = e^{i*2\pi*f*t/N} \quad (3)$$

The corresponding frequency is:

$$f_i = f * n_i / N \quad (4)$$

With the total number of data $N = 10^5$, we know that, the frequency of the instrument is:

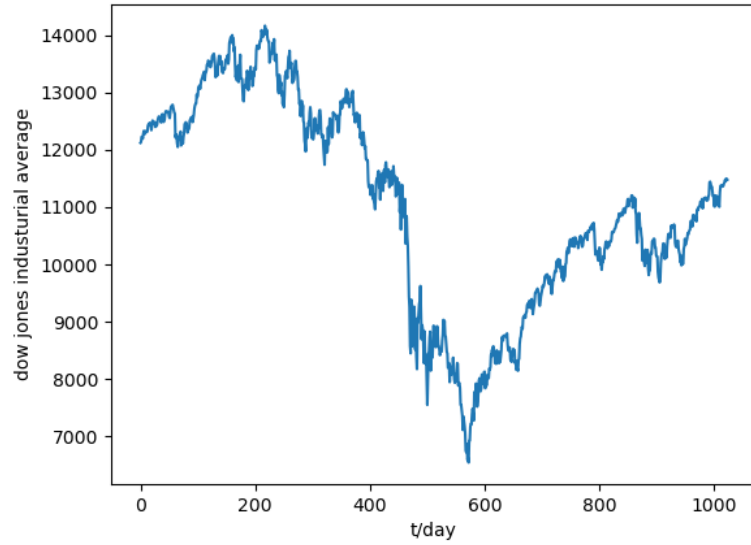
- frequency for piano is 524.79Hz
- frequency for trumpet is 522.14Hz

We conclude that the musical note it's playing is C5.

4 Newman 7.4: Fourier filtering and smoothing

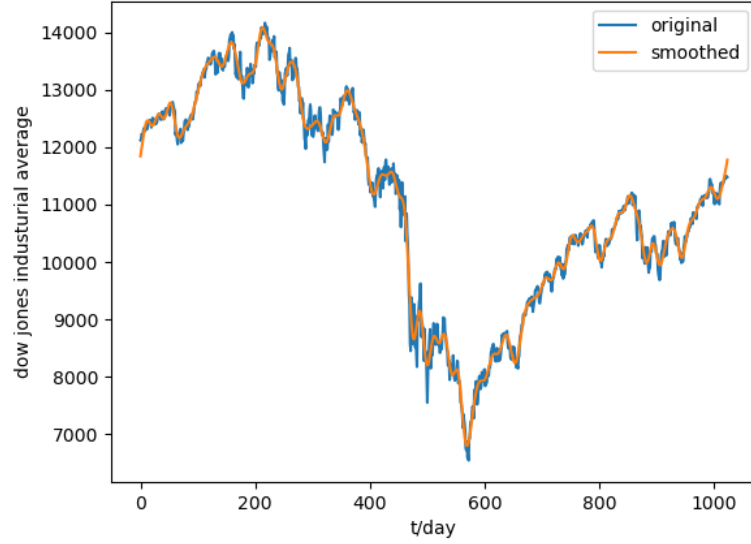
We use numpy to read the txt file into numpy array. This is plotted as follows:

Table 8: Dow jones industrial index over time



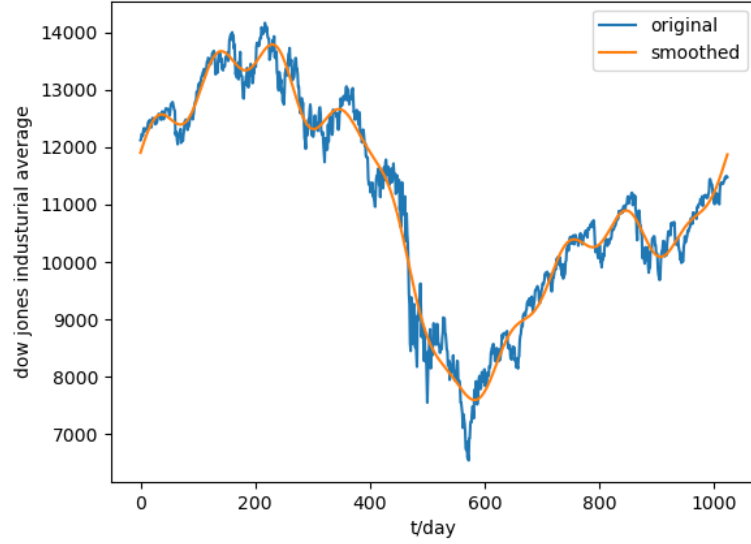
We can then do the FFT. We keep the first 10% of fourier components and do the inverse FFT. The result is plotted below:

Table 9: Dow jones(original and 10% smoothed) industrial index over time



We can then do the same thing while only keep first 2% of the fourier components. The result is plotted below:

Table 10: Dow jones(original and 2% smoothed) industrial index over time



We can clearly see that, when we only keep low frequency fourier components, the real curve is smoothed with less short-range detail. The less fourier components we keep, the more the original function is smoothed.