Лабораторная работа №5
Автокорреляция

Крынский Павел

26 мая 2021 г.

Оглавление

Список иллюстраций

Листинги

Глава 1

Упражнение 5.1

Я позаимствовал запись вокального чирпа, который использовался выше.

```
wave = thinkdsp.read_wave('28042__bcjordan__voicedownbew.wav')
wave.normalize()
wave.make_audio()
```

Листинг 1.1: Вокальный чирп

Далее я взял сегмент, который начинается с 0.3 секунды после начала и длительностью 0.01 секунды:

```
sg = wave.segment(start = 0.2 , duration = 0.01)
```

Листинг 1.2: Первый сегмент

Применим автокорреляционную функцию, чтобы оценить высоту тона:

```
lags, corrs = autocorr(sg)
thinkplot.plot(lags, corrs)
thinkdsp.decorate(xlabel='Lag(index)',ylabel='Correlation')
```

Листинг 1.3: Оценка высоты тона

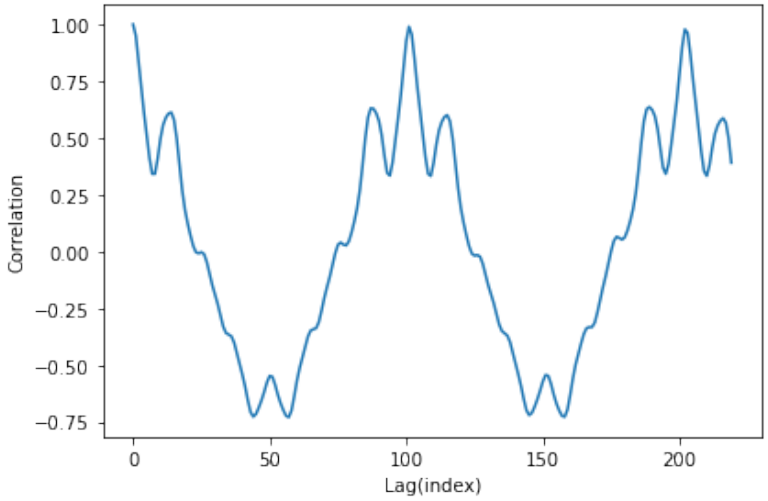

Рис. 1.1: Высота тона

Пик находится между 100 и 150. Используем `argmax`, чтобы уточнить значение `lag` для этого пика:

```
lag = np.array(corrs[100:150]).argmax() + 100
lag
```

Листинг 1.4: Нахождение `lag`

Находим соответствующую частоту для `lag` = 109:

```
p = lag / sg.framerate
fr = 1/p
fr
```

Листинг 1.5: Нахождение частоты

Частота равняется `436.63366336633663`.

Теперь рассмотрим сегмент сигнала через 1 секунду и проделаем аналогичные действия.

```
sg = wave.segment(start = 1 , duration = 0.01)
```

Листинг 1.6: Второй сегмент

Применим автокорреляционную функцию, чтобы оценить высоту тона:

```
lags, corrs = autocorr(sg)
thinkplot.plot(lags, corrs)
```

```
thinkdsp.decorate(xlabel='Lag(index)',ylabel='Correlation')
```

Листинг 1.7: Оценка высоты тона

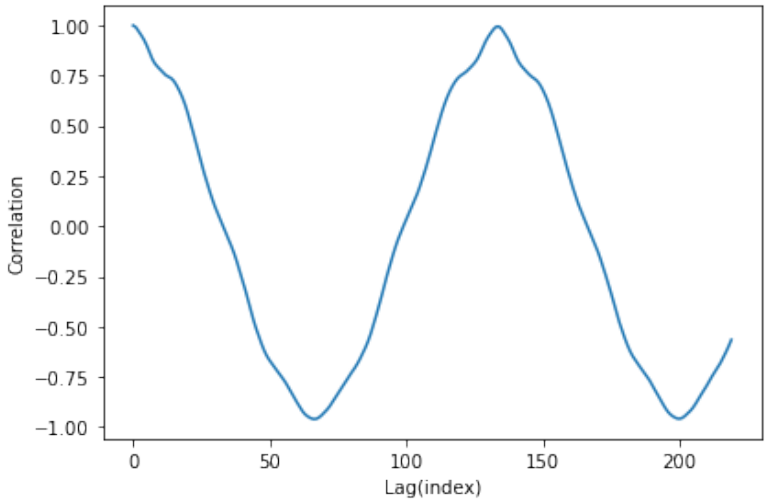

Рис. 1.2: Высота тона

Пик снова находится между 100 и 150. Используем `argmax`, чтобы уточнить значение `lag` для этого пика:

```
lag = np.array(corrs[100:150]).argmax() + 100
lag
```

Листинг 1.8: Нахождение `lag`

Находим соответствующую частоту для `lag` = 134:

```
p = lag / sg.framerate
fr = 1/p
fr
```

Листинг 1.9: Нахождение частоты

Частота равняется `329.1044776119403`. Отсюда можно сделать вывод, что основная частота ожидаемо уменьшается при увеличении времени начала сегмента.

Глава 2

Упражнение 5.2

Загрузим тот же вокальный чирп.

```
wave = thinkdsp.read_wave('28042__bcjordan__voicedownbew.wav')
wave.normalize()
wave.make_audio()
```

Листинг 2.1: Загрузка звука

Воспользуемся примером кода из `chap05.ipynb`. Вот его спектрограмма:

```
wave.make_spectrogram(2048).plot(high=4200)
thinkdsp.decorate(xlabel='Time (s)',ylabel='Frequency (Hz)')
```

Листинг 2.2: Спектрограмма звука

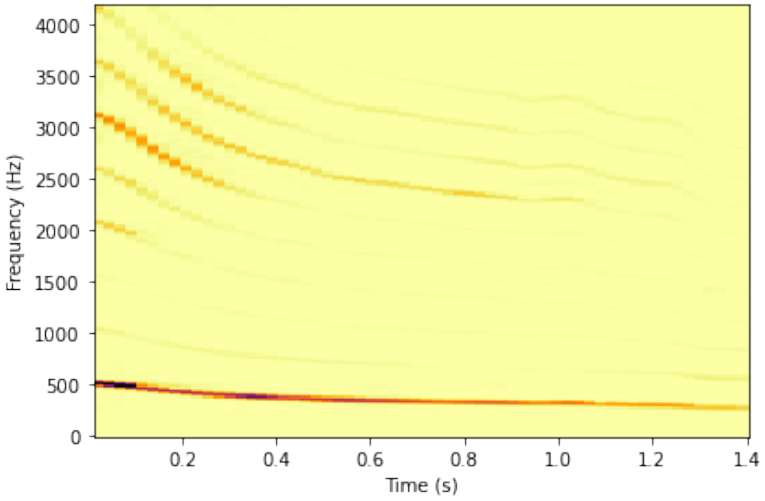

Рис. 2.1: Сегменты звуков

Инкапсулируем предлагаемый код в функцию. Найти первый самый высокий пик в автокорреляционной функции сложно. Поэтому мы просто укажем диапазон `lag` для поиска.

```
def estimate_fundamental(segment, low=70, high=150):
    lags, corrs = autocorr(segment)
    lag = np.array(corrs[low:high]).argmax() + low
    period = lag / segment.framerate
    frequency = 1 / period
    return frequency
```

Листинг 2.3: Инкапсуляция функции

Рассмотрим пример работы функции:

```
duration = 0.01
segment = wave.segment(start=0.2, duration=duration)
freq = estimate_fundamental(segment)
freq
```

Листинг 2.4: Пример работы функции

Результатом работы получилась частота `436.63366336633663`.

Используем написанную функцию для отслеживания высоты тона записанного звука. `ts` - средние точки каждого сегмента.

```
step = 0.05
starts = np.arange(0.0, 1.4, step)

```

```
ts = []
freqs = []

for start in starts:
    ts.append(start + step/2)
    segment = wave.segment(start=start, duration=duration)
    freq = estimate_fundamental(segment)
    freqs.append(freq)
```

Листинг 2.5: Отслеживание высоты тона

Рассмотрим кривую отслеживания высоты тона, наложенную на спектрограмму:

```
wave.make_spectrogram(2048).plot(high=900)
thinkplot.plot(ts, freqs, color='blue')
thinkdsp.decorate(xlabel='Time (s)', ylabel='Frequency (Hz)')
```

Листинг 2.6: Кривая отслеживания высоты тона на спектрограмме

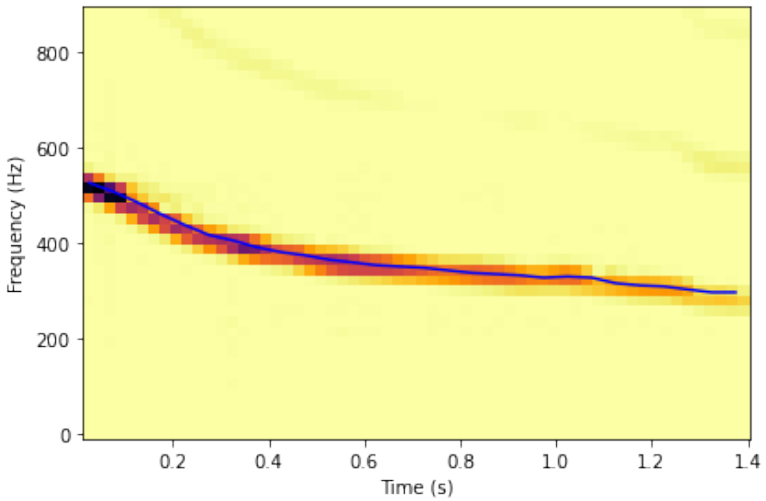

Рис. 2.2: Кривая отслеживания высоты тона на спектрограмме

Наложив оценки высоты тона на спектрограмму записи, можно видеть, что функция полностью справляется со своей задачей.

Глава 3

Упражнение 5.3

Воспользуемся данными о ежедневной цене BitCoin в течение года из прошлой лабораторной работы.

```
data = pd.read_csv('BTC_USD_2013-10-01_2021-05-22-CoinDesk.csv',
    parse_dates=[0])
data
```

Листинг 3.1: Таблица данных

Визуализируем скачанные данные.

```
wave = thinkdsp.Wave(data['Closing Price (USD)'], data.index,
    framerate=1)
wave.plot()
thinkdsp.decorate(xlabel='Time (days)', ylabel='Price of BitCoin
    ($)')
```

Листинг 3.2: Визуализация данных

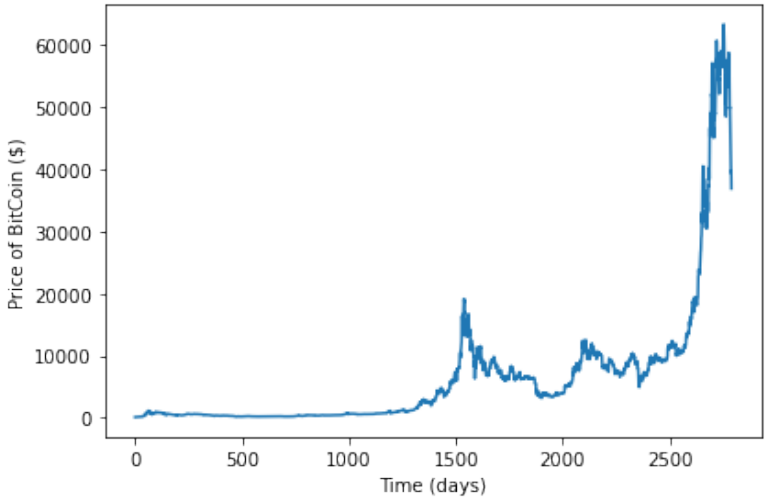

Рис. 3.1: Визуализация данных

Воспользуемся функцией автокорреляции, использующая статистическое определение, то есть она сдвигает среднее значение к нулю, делит на стандартное отклонение и делит сумму на N.

```
from autocorr import autocorr

lags, corrs = autocorr(wave)
thinkplot.plot(lags, corrs)
thinkdsp.decorate(xlabel='Lag', ylabel='Correlation')
```

Листинг 3.3: Автокорреляция при помощи `autocorr`

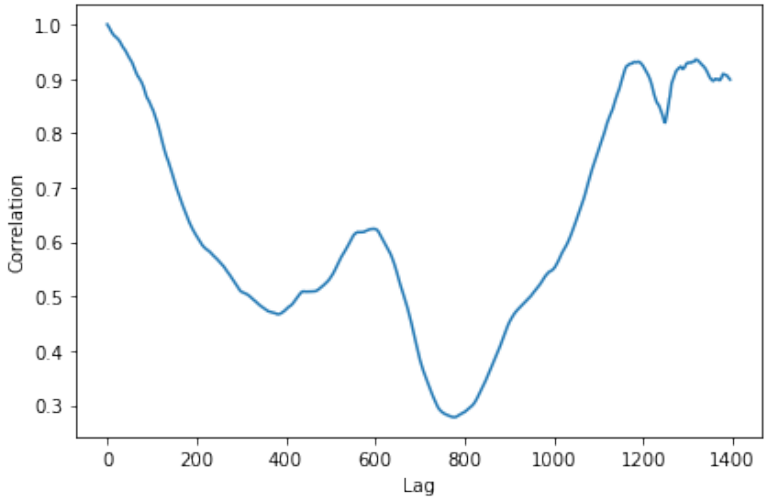

Рис. 3.2: Автокорреляция при помощи `autocorr`

Автокорреляционная функция падает медленно, поскольку задержка увеличивается, предполагая какой-то розовый шум.

Результат симметричен, потому что два сигнала идентичны, и отрицательный `lag` у одного даёт такой же эффект, как и положительный `lag` у другого. Вторая половина результата соответствует положительным `lag`:

```
N = len(wave)
corrs2 = np.correlate(wave.ys, wave.ys, mode='same')
lags = np.arange(-N//2, N//2)
thinkplot.plot(lags, corrs2)
thinkdsp.decorate(xlabel='Lag',ylabel='Dot product')
```

Листинг 3.4: Вторая половина результата

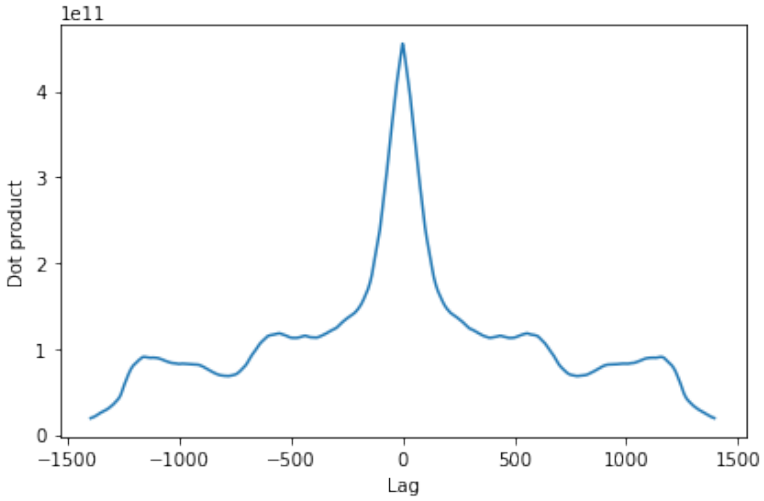

Рис. 3.3: Автокорреляция при помощи `autocorr`

Для этого набора данных, вероятно, более уместно статистическое определение автокорреляционной функции.

Глава 4

Упражнение 5.4

В данном упражнении я просмотрел прикрепленный ролик на YouTube и изучил все примеры, которые были приведены в файле saxophone.ipynb для разных форматов записи.

Глава 5

Выводы

Во время выполнения лабораторной работы получены навыки работы с корреляцией, последовательной корреляцей, автокорреляцей. Также рассмотрена автокорреляционная функция (АКФ).