Jacob T. VanderPlas Understanding the Lomb–Scargle Periodogram // The Astrophysical Journal Supplement Series, 236:16 (28pp), 2018, May, http://dx.doi.org/10.3847/1538-4365/aab766

# Lomb-Scargle метод

Периодограмма Ломб-Скаргла (Lomb 1976; Scargle 1982) - это хорошо известный алгоритм обнаружения и определения периодичности в неравномерно отсортированных временных рядах, отличающийся особенно широким применением в астрономическом сообществе.

Часто именно так и представляется периодограмма Ломба-Скаргла: как чистая, четко определенная процедура для генерации спектра мощности и обнаружения периодического компонента в неравномерно отобранном наборе данных. Однако на практике существует ряд тонких вопросов, которые необходимо учитывать при применении анализа Lomb-Scargle к наборам данных в реальном мире. В частности, вот несколько вопросов, которые можно задать:

1. Как соотносится периодограмма Ломба-Скаргла с классическим Фурье-спектром?

2. Каков источник множественных пиков, выявляемых периодограммой Ломба-Скаргла?

3. Какую самую большую частоту (Найквиста) воспроизводит такой анализ?

4. Как следует выбирать интервал частотной сетки для нашей периодограммы?

5. Какие предположения о форме неизвестного сигнала делается в методе Ломб-Скаргла?

6. Какова неопределенность рассчитанной частоты?

## Пример.

Рассмотрим тестовый пример синусоиды со случайными пропусками в данных.

**import** **matplotlib.pyplot** **as** **plt**

**from** **matplotlib** **import** cm

**import** **scipy.signal** **as** **signal**

**import** **numpy** **as** **np**

%**matplotlib** inline

*#First define some input parameters for the signal:*

*#*

A = 2.

w = 1.

phi = 0.5 \* np.pi

nin = 1000

nout = 1000

frac\_points = 0.95 *# Fraction of points to select*

*#Randomly select a fraction of an array with timesteps:*

r = np.random.rand(nin)

xt = np.linspace(0.01, 10\*np.pi, nin)

*#print(len(x))*

*#dx=x[1]-x[0]*

dx=(10.\*np.pi-0.01)/(nin-1)

x = xt[r >= frac\_points]

*#Plot a sine wave for the selected times:*

y = A \* np.sin(w\*x+phi)

*# Define the array of frequencies for which to compute the periodogram:*

min\_fraq=1./x[len(x)-1]

max\_fraq=1./(2.\*dx)

*#f = np.linspace(0.01, 4., nout)*

f = np.linspace(min\_fraq, max\_fraq, nout)

*#Calculate Lomb-Scargle periodogram:*

*#pgram = signal.lombscargle(x, y, f, normalize=False)*

pgram = signal.lombscargle(x, y, f)*#,normalize=True)*

*#Now make a plot of the input data:*

plt.subplot(2, 1, 1)

plt.plot(x, y, 'b+')

*#Then plot the periodogram:*

plt.subplot(2, 1, 2)

plt.plot(f, pgram)

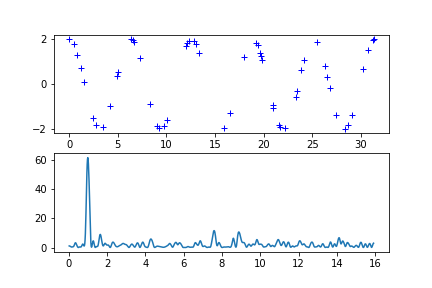


Рис. 1. Ряд с пропусками (вверху) и его периодограмма (внизу). По оси Х нижнего рисунка отложены частоты.

Как видим, наряду с основным (и единственным в исходных данных) максимумом, появились небольшие дополнительные, которые можно не учитывать, отсекая их по некоторому правилу.

Для анализа длинного временного ряда зачастую надо знать, как меняется спектр с течением времени. Один из стандартных способов – применение оконного преобразования, в нашем случае Ломб-Скаргл оконного преобразования. Качество полученных результатов будет зависеть от количества пропусков данных и ширине окна. В зависимости от соотношения количества число\_пропусков/число\_измерений следует выбирать ширину окна. На рисунке 2 представлены оконные спектры тестового сигнала, приведенного на Рис. 1 с различными размерами «окна», равными одному периоду, двум периодам и трем периодам соответственно.

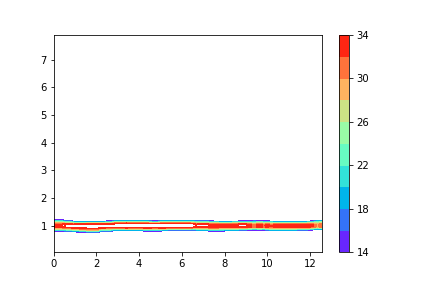
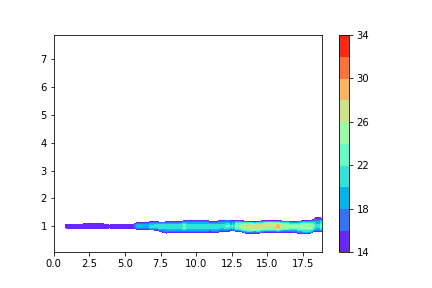
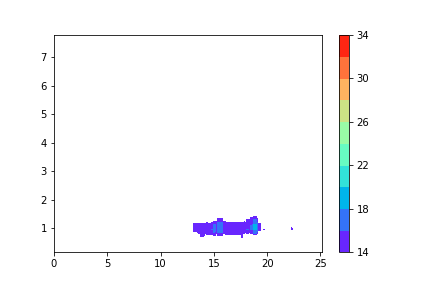


Рис. 2. Оконные LS периодограммы тестового сигнала для различного размера «окна» (сверху вниз): 1Т, 2Т и 3Т соответственно.