УДК 004.624

Оценка плотности ядра на языке программирования Python

Кизянов Антон Олегович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье будет показано, как на примере 4 библиотек можно провести оценку плотности ядра на языке программирования Python
Ключевые слова: Python, SciPy, statsmodels, scikit-learn, Seaborn

Estimating kernel density in the Python programming language

Kizyanov Anton Olegovich
Sholom-Aleichem Priamursky State University
student

Abstract
This article will tell you how, as an example of 4 libraries, you can evaluate the kernel density in the Python programming language.
Keywords: Python, SciPy, statsmodels, scikit-learn, Seaborn

Часто есть предположение о том, какое распределение больше подходит для наших данных. Если нет, можно применить процедуру, называемую оценкой плотности ядра. Он в основном сглаживает данные, пытаясь получить дескриптор плотности вероятности. Чтобы сгладить данные, можно использовать различные функции. Эти функции называются функциями ядра. Следующее уравнение определяет общий вид формулы:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i-1}^{n} K(\frac{x - x_i}{h}) \qquad (1)$$

В предыдущей формуле K - функция ядра, функция со свойствами, подобными функции распределения вероятности. Параметр полосы пропускания h контролирует процесс сглаживания и может сохраняться фиксированным или изменяющимся. Некоторые библиотеки используют эмпирические правила для вычисления h, в то время как другие позволяют указать его значение. SciPy, statsmodels, scikit-learn и Seaborn осуществляют оценку плотности ядра с использованием разных алгоритмов.

Цель исследования – демонстрация оценки плотности ядра с помощью четырех разных библиотек (SciPy, statsmodels, scikit-learn и Seaborn).

Ранее этим вопросом интересовались А.В. Петрухин, А.С. Стешенко развивали тему «Компьютерная визуализация биржевых данных о динамике

фондового рынка» [1] в которой рассказывается про типы формализованных отображений информации, получаемой с биржи посредством торговых терминалов. Показаны проблемы, возникающие при построении специализированных модулей биржевой визуализации. Предлагаются методики построения подсистем визуализации с использованием библиотек OxyPlot и matplotlib, упрощающие процесс построения соответствующих программных приложений. Д.А. Хворостов опубликовал статью «Применение профессиональных компьютерных технологий в проектной деятельности студентов-дизайнеров» [2] рассказал про особенности применения профессиональных компьютерных технологий в проектной деятельности студентов-дизайнеров. Ф.М. Коркмасов опубликовал статью «Математическая обработка данных геотермальных исследований и терморазведки» [3] рассказал про регистрацию радиотеплового и инфракрасного излучения земной поверхности, измерение температуры, теплового потока и распределение этих параметров в плане и по глубине позволяют получить информацию о термических условиях и геологическом строении изучаемого района.

Сначала нужно импортировать все нужные библиотеки.

```
import seaborn as sns
import matplotlib.pyplot as plt
import dautil as dl
from dautil.stats import zscores
import statsmodels.api as sm
from sklearn.neighbors import KernelDensity
import numpy as np
from scipy import stats
from IPython.html import widgets
from IPython.core.display import display
from IPython.display import HTML
```

Определить функцию для построения расчетной плотности ядра:

```
def plot(ax, a, b, c, xlabel, ylabel):
    dl.plotting.scatter_with_bar(ax, 'Kernel Density', a.values, b.values, c=c,
cmap='Blues')
    ax.set_xlabel(xlabel)
    ax.set_ylabel(ylabel)
```

Загрузить данные и определить виджеты для выбора переменных:

```
df = dl.data.Weather.load().resample('M').dropna()
columns = [str(c) for c in df.columns.values]
var1 = widgets.Dropdown(options=columns, selected_label='RAIN')
display(var1)
var2 = widgets.Dropdown(options=columns, selected_label='TEMP')
display(var2)
```

Определить переменные, используя значения созданных виджетов:

```
x = df[var1.value]
xlabel = dl.data.Weather.get_header(var1.value)
y = df[var2.value]
ylabel = dl.data.Weather.get_header(var2.value)
X = [x, y]
```

Следующий код обрисовывает важные линии каждого графика:

```
Z = [zscores(x), zscores(y)]
kde = stats.gaussian_kde(Z)
_, [[sp_ax, sm_ax], [sk_ax, sns_ax]] = plt.subplots(2, 2)
plot(sp_ax, x, y, kde.pdf(Z), xlabel, ylabel)
sp_ax.set_title('SciPy')

sm_kde = sm.nonparametric.KDEMultivariate(data=X, var_type='cc',
                                          bw='normal_reference')
sm_ax.set_title('statsmodels')
plot(sm_ax, x, y, sm_kde.pdf(X), xlabel, ylabel)
XT = np.array(X).T
sk_kde = KernelDensity(kernel='gaussian', bandwidth=0.2).fit(XT)
sk_ax.set_title('Scikit Learn')
plot(sk_ax, x, y, sk_kde.score_samples(XT), xlabel, ylabel)
sns_ax.set_title('Seaborn')
sns.kdeplot(x, y, ax=sns_ax)
sns.rugplot(x, color="b", ax=sns_ax)
sns.rugplot(y, vertical=True, ax=sns_ax)
sns_ax.set_xlabel(xlabel)
sns_ax.set_ylabel(ylabel)
plt.tight_layout()
HTML(dl.report.HTMLBuilder().watermark())
```

Результат работы представлен на рисунке 1.

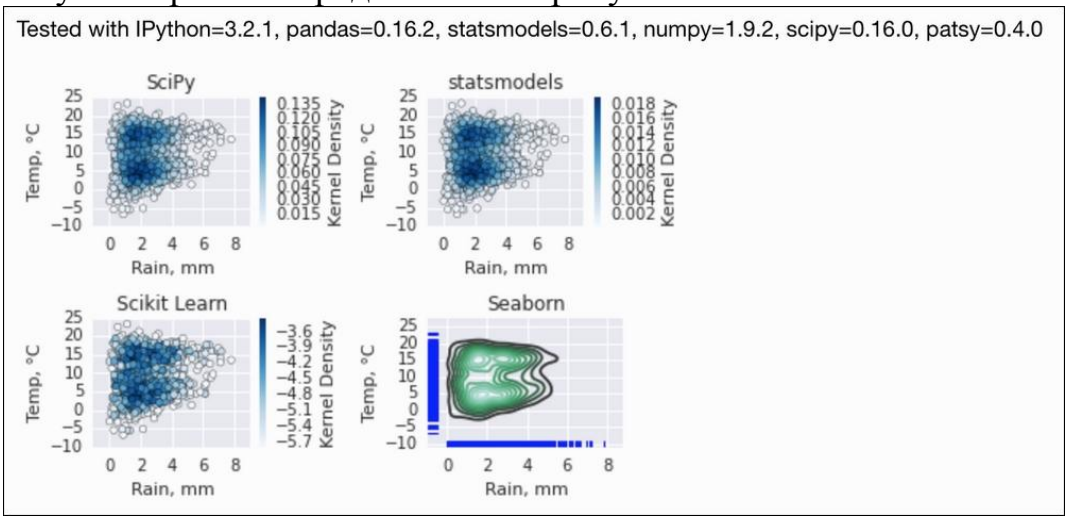

Рис. 1 Четыре графика демонстрирующих плотность ядра в разных библиотеках

Вывод

Таким образом работать с данными можно в любой из используемых библиотек, результат примерно одинаков, за исключением библиотеки Seaborn, в ней используется не точный, а рельефный график. Выбор библиотеки для работы больше зависит от собственных предпочтений и требований проекта.

Библиографический список

1. Петрухин А.В., Стешенко А.С. Компьютерная визуализация биржевых данных о динамике фондового рынка // 2015. № 6 (163). С. 124-129. URL: https://elibrary.ru/item.asp?id=24334292 (Дата обращения: 09.08.2018)
2. Хворостов Д.А. Применение профессиональных компьютерных технологий в проектной деятельности студентов-дизайнеров // 2017. № 2. С. 105-109. URL: https://elibrary.ru/item.asp?id=32287436 (Дата обращения: 09.08.2018)
3. Коркмасов Ф.М. Математическая обработка данных геотермальных исследований и терморазведки // Возобновляемая энергетика: проблемы и перспективы. 2005. С. 288-294. URL: https://elibrary.ru/item.asp?id=21550740 (Дата обращения: 09.08.2018)