Статистический анализ изменения частоты сенсоров

Оглавление

1. Проверка нормальности распределения векторов dF

Большинство статистических методов анализа данных исходят из предположения о том, что данные имеют нормальное распределение, следовательно необходимо начать с критерия нормальности. В пакете научных вычислений для Python Scipy [1] реализован [2] обобщённый тест Д'Агостино-Пирсона на нормальность распределения [3]. Графики построены с помощью графика квантилей, реализованного в виде функции qqplot [4] в пакете прикладной статистики statsmodels [5]. Графики имеют S-образную форму, что говорит о том, что не все данные соответствуют нормальному распределению.

Чтобы окончательно убедиться в этом, выполним тест Харке-Бера, сохраним результаты в Pandas DataFrame а затем установим порог вероятности равным 5%: если значение столбца p-value будет превышать этот порог, то значит, что нулевая гипотеза о нормальности распределения отвергается для указанного временного ряда. С помощью функции norm_analisys выясним, что в собранном массиве данных Pandas существует 48 рядов, распределение которых не соответствует нормальному. Видно, что таковыми являются не целые вещества, а показания отдельных сенсоров.

Код: `arima.py`, функция `norm_test`

Выход: `graphs/norm` – содержит графики квантилей

Код: `arima.py`, функция `jarque_bera_test`

Выход: `jarque_bera.txt` – содержит таблицу с выводом результатов теста Харке-Бера

Код: `arima.py`, функция `jarque_bera_test_analisys`

Выход: в консоль – количество векторов, чье распределение не соответствует нормальному.

2. Проверка на стационарность векторов dF

Начнем с вычисления скользящих статистик (среднего и стандартного отклонения) и теста на единичные корни. В пакете statsmodels реализован расширенный тест Дики-Фуллера (ADF — Augmented Dickey-Fuller) [6]. Его преимущество перед обычным тестом Дики-Фуллера (DF) состоит в том, что, благодаря включению первых разностей, появляется возможность работать с авторегрессиями не только первого, но и более высоких порядков, так как авторегрессия еще не была исследована. В результате обнаружено, что существует 228 нестационарных рядов из 288 — то есть можно сделать вывод, что большинство из них нестационарны.

Код: `arima.py`, функция `test_stationarity`

Выход: `graphs/stat` – содержит графики скользящих статистик (мат.ожидания и стандартного отклонения) и аутпут теста Дики-Фуллера

Код: `arima.py`, функция `a_dickey_fully_test`

Выход: adf_protocol.txt – содержит таблицу с результатами (единичный корень и стаицонарность)

Код: `arima.py`, функция `a_dickey_fully_test_analisys`

Выход: в консоль – количество нестационарных векторов

3. Анализ автокорреляционных и частичных автокорреляционных функций

Автокорреляционная функция негладкая для: ацетальдегида 0 (S7), ацетальдегида 3 (S2, S3, S7), ацетон 6 (S1, S7), бензин (S2, S7), диоктилфталат 15 (S1), диоктилфталат 16 (S7), диоктилфталат 17 (S7), диоктилфталат 18 (S1, S7), диоктилфталат 20 (S7), диоктилфталат 21 (S2, S4, S7, S8), диоктилфталат 22 (S1, S4, S7, S8), диоктилфталат 23 (S7), этилацетат 32 (S1), этилацетат 33

(S2), этилацетат 35 (S2), гексан 14 (почти весь), пластизоль 26 (S2, S4, S5), пластизоль 27 (S3, S4, S7) — всего 32 вектора.

В дальнейшей работе планирую вернуться анализу результатов.

Код: `arima.py`, функция `autocorr`

Выход: `graphs/auto` – содержит графики автокорреляции и частичной автокорреляции

4. Авторегрессионные модели

Поскольку не все векторы стационарны, целесообразно остановить свой выбор на модели Бокса-Дженсинса [7] или ARIMA (Autoregressive Integrated Moving Average) [8]. Модель, реализованая в пакете statsmodels, предлагает удобный интерфейс как для задания параметров модели [9], так и для применения этой модели к данным [10]. Функция fit() позволяет настроить такие глобальные параметры: “на лету” приводить исследуемый ряд к стационарному (*transparams*), выбрать один из трех методов работы с максимальным правдоподобием (*method*), учет тренда (*trend*), солвер (*solver*), количество итераций (*maxiter*).

Текущие глобальные параметры: transparams=True (defalut), так как не все исследуемые временные ряды стационарны, method=css-mle (default) – условный метод максимального правдопдобия, trend=c (default) – учитывать константу, solver=newton -

Подбор параметров p, d, q будет осуществляться автоматически из следующих интервалов: $p \in [0,3], d \in [0,2], q \in [0,3]$, поскольку более интеллектуальный выбор параметров затрудняется большим количеством векторов (8 векторов из 36 матриц веществ – 288 векторов), поведение которых отличается. В качестве критерия отбора выбрана минимизация информационного критерия Акаике.

Дальнейшие эксперименты будут проводиться с увеличением порядка модели.

Код: `arima.py`, функция `arima_find_best`

Выход: `arima_est.txt`, содержит таблицу с наилучшими параметрами.

Ссылки

1. https://www.scipy.org/

2. https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.normaltest.html

3. D'Agostino, R. B. (1971), "An omnibus test of normality for moderate and large sample size," Biometrika, 58, 341-348

4. http://statsmodels.sourceforge.net/devel/generated/statsmodels.graphics.gofplots.qqplot.html

5. http://statsmodels.sourceforge.net/

6. http://statsmodels.sourceforge.net/devel/generated/statsmodels.tsa.stattools.adfuller.html

7. Box, G.E.P., and Jenkins, G., (1970) Time Series Analysis, Forecasting and Control, Holden-Day, San Francisco.

8. Box, G.E.P., and Pierce, D.A., (1970) "Distribution of the Residual Autocorrelations in Autoregressive-Integrated Moving-Average Time Series Models", Journal of the American Statistical Association, 65, 1509-1526.

9. http://statsmodels.sourceforge.net/devel/generated/statsmodels.tsa.arima_model.ARIMA.html

10. http://statsmodels.sourceforge.net/devel/generated/statsmodels.tsa.arima_model.ARIMA.fit.html