Федеральное государственное автономное
образовательное учреждение высшего образования
«Омский государственный технический университет»

Кафедра «Информатика и вычислительная техника»

Расчетно-графическая работа

по дисциплине «Интеллектуальные системы»
на тему «Решение задач машинного обучения на примере модели Тейла-Сена»

Выполнила:
студентка группы ЗИВТм-231
________ Кузнецова Д. А.

Проверил:
ст. преподаватель
________ Убалехт И. П.

Омск 2024

Содержание

Введение ... 3

1. Описание модели Тейла-Сена ... 4

2. Доказательства устойчивости модели Тейла-Сена к шуму в данных . 6

3. Демонстрация работы метода ... 7

Заключение ... 9

Список использованной литературы ... 10

Введение

Регрессия - это статистический метод, используемый для изучения отношений между зависимой переменной (которую мы хотим предсказать) и независимыми переменными (которые используются для предсказания).

Регрессия позволяет определить, как независимые переменные влияют на зависимую переменную и создать модель, которая может предсказывать значения зависимой переменной на основе значений независимых переменных.

В данной расчетно-графической работе были рассмотрены три вида регрессий: OLS (обычный метод наименьших квадратов), Theil-Sen и RANSAC.

Метод обычных наименьших квадратов (OLS) является классическим методом линейной регрессии, который минимизирует сумму квадратов остатков между наблюдаемыми значениями зависимой переменной и значениями, предсказанными моделью.

Theil-Sen регрессия также относится к линейной регрессии, но вместо минимизации суммы квадратов остатков, она использует медианную наклонную оценку для определения наклона линии регрессии. Этот метод более устойчив к выбросам в данных.

RANSAC (Random Sample Consensus) также является методом линейной регрессии, который пытается найти линию регрессии, игнорируя выбросы в данных. Он достигается путем случайной выборки подмножества точек данных и нахождения наилучшей модели, которая соответствует этим точкам.

В качестве объекта изучения и модификации был выбран код на языке Python, который вычисляет регрессию Тейла-Сена на синтетическом наборе данных.

Целью данной расчетно-графической работы является изучение влияния шума на зависимую и независимую переменные при построении регрессии Тейла-Сена и сравнение этой модели регрессии с другими моделями.

1. Описание модели Тейла-Сена

В данной работе был использован код на языке программирования Python. Код демонстрирует преимущество модели Тейла-Сена при применении различных методов регрессии к данным с аномалиями. В данном случае, были использованы методы обычной линейной регрессии (OLS), модель Тейла-Сена и RANSAC.

Этапы создания модели:

1. Сначала создаются данные для моделирования. Для этого генерируются случайные значения x и добавляется шум к зависимой переменной y.

2. Затем создается модель с выбросами в данных, где 10% значений были изменены, чтобы создать аномалии.

3. Далее для каждого метода регрессии (OLS, модель Тейла-Сена, RANSAC) производится обучение модели на подготовленных данных.

4. После обучения модели предсказания делаются для новых значений x, и результаты визуализируются на графиках, где показаны исходные данные (точки) и предсказания моделей (линии) для каждого метода регрессии.

По сравнению с оценкой OLS (метод наименьших квадратов), модель Тейла-Сена устойчива к выбросам. Она имеет точку разрушения около 29,3% в случае простой линейной регрессии, что означает, что она может допускать произвольные искаженные данные (выбросы) до 29,3% в двумерном случае. Оценка модели выполняется путем расчета наклонов и точек пересечения подгруппы всех возможных комбинаций p точек подвыборки. Если установлен перехват, p должен быть больше или равен n_features + 1. Окончательный наклон и перехват затем определяется как пространственная медиана этих наклонов и перехватов. В некоторых случаях модель Тейла-

Сена работает лучше, чем RANSAC, который также является надежным методом.

На рис.1 представлен демонстрационный код, который показывает, как различные методы регрессии могут обрабатывать данные с аномалиями.

```
import time
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression, RANSACRegressor, TheilSenRegressor
estimators = [
("OLS", LinearRegression()),
("Theil-Sen", TheilSenRegressor(random_state=42)),
("RANSAC", RANSACRegressor(random_state=42)),
]
colors = {"OLS": "turquoise", "Theil-Sen": "gold", "RANSAC": "lightgreen"}
np.random.seed(0)
lw = 2 * np.random.rand()
n_samples = 200

# Linear model y = 3 * x + N(2, 0.1**2)
x = np.random.randn(n_samples)
w = 3.0
c = 2.0
noise = 0.1 * np.random.randn(n_samples)
y = w * x + c + noise

# 10% outliers
y[-20:] += -20 * x[-20:]
X = x[:, np.newaxis]
plt.scatter(x, y, color="indigo", marker="x", s=40)
line_x = np.array([-3, 3])
for name, estimator in estimators:
    t0 = time.time()
    estimator.fit(X, y)
    elapsed_time = time.time() - t0
    y_pred = estimator.predict(line_x.reshape(2, 1))
    plt.plot(
        line_x,
        y_pred,
        color=colors[name],
        linewidth=lw,
        label="%s (fit time: %.2fs)" % (name, elapsed_time),
   )
plt.axis("tight")
plt.legend(loc="upper left")
plt.title("Corrupt y")
np.random.seed(0)

# Linear model y = 3*x + N(2, 0.1**2)
x = np.random.randn(n_samples)
noise = 0.1 * np.random.randn(n_samples)
y = 3 * x + 2 + noise


x[-20:] = 9.9
y[-20:] += 22
X = x[:, np.newaxis]
plt.figure()
plt.scatter(x, y, color="indigo", marker="x", s=40)
line_x = np.array([-3, 10])
for name, estimator in estimators:
    t0 = time.time()
    estimator.fit(X, y)
    elapsed_time = time.time() - t0
    y_pred = estimator.predict(line_x.reshape(2, 1))
    plt.plot(
        line_x,
        y_pred,
        color=colors[name],
        linewidth=lw,
        label="%s (fit time: %.2fs)" % (name, elapsed_time),
    )
plt.axis("tight")
plt.legend(loc="upper left")
plt.title("Corrupt x")
plt.show()
```

Рисунок 1 - Описание модели Тейла-Сена

2. Доказательства устойчивости модели Тейла-Сена к шуму в данных

Для изучения влияния шума на зависимую и независимую переменные при построении регрессии Тейла-Сена в исходный код был добавлен дополнительный шум к данным.

На рис.2 представлен код после внесенных изменений.

Edit View Insert Cell Kernel Widgets Help

Run Code

```
[31]: import time
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression, RANSACRegressor, Thei

estimators = [
    ("OLS", LinearRegression()),
    ("Theil-Sen", TheilSenRegressor(random_state=42)),
    ("RANSAC", RANSACRegressor(random_state=42)),
]

colors = {"OLS": "turquoise", "Theil-Sen": "gold", "RANSAC": "lightgreer
np.random.seed(0)
plt.scatter(x, y, color="indigo", marker="x", s=40)
line_x = np.array([-3, 3])

for name, estimator in estimators:
    t0 = time.time()
    estimator.fit(X, y)
    elapsed_time = time.time() - t0
    y_pred = estimator.predict(line_x.reshape(2, 1))
    plt.plot(
        line_x,
        y_pred,
        color=colors[name],
        linewidth=lw,
        label="%s (fit time: %.2fs)" % (name, elapsed_time),
    )
plt.axis("tight")
plt.legend(loc="upper left")
plt.title("Corrupted y with added noise")
np.random.seed(0)

x = np.random.randn(n_samples)
noise = 0.1 * np.random.randn(n_samples)
y = 3 * x + 2 + noise

x[-20:] = 9.9
y[-20:] += 22

x += np.random.normal(scale=0.5, size=n_samples) # Добавление шума к данным
y += np.random.normal(scale=0.5, size=n_samples) # Добавление шума к данным
X = x[:, np.newaxis]
plt.figure()
plt.scatter(x, y, color="indigo", marker="x", s=40)
line_x = np.array([-3, 10])

for name, estimator in estimators:
    t0 = time.time()
    estimator.fit(X, y)
    elapsed_time = time.time() - t0
    y_pred = estimator.predict(line_x.reshape(2, 1))
    plt.plot(
        line_x,
        y_pred,
        color=colors[name],
        linewidth=lw,
        label="%s (fit time: %.2fs)" % (name, elapsed_time),
    )
plt.axis("tight")
plt.legend(loc="upper left")
plt.title("Corrupted x and y with added noise")
plt.show()
```

Рисунок 2 – Внесение дополнительного шума в переменные

Код до внесенных изменений использует данные x для генерации зависимости y = w × x + c + noise, где w = 3,0, c = 2,0, и добавляет шум к последним 20 точкам данных. Затем он использует обученные модели для предсказания значений y на прямой линии line_x и визуализирует результаты.

Измененный код также использует данные x для генерации зависимости y = w × x + c + noise, но затем он добавляет дополнительный шум к значениям y и x. Таким образом, второй код демонстрирует влияние добавления дополнительного шума как к зависимой переменной y, так и к независимой переменной x.

Добавление дополнительного шума к данным может быть полезным для демонстрации влияния шума на процесс обучения моделей. Это позволяет показать, как модели реагируют на различные уровни шума и какие компоненты данных оказывают наибольшее влияние на точность предсказаний.

3. Демонстрация работы метода

На рис. 3 изображены графики до внесения изменений в исходный код, а на рис. 4 после внесенных изменений.

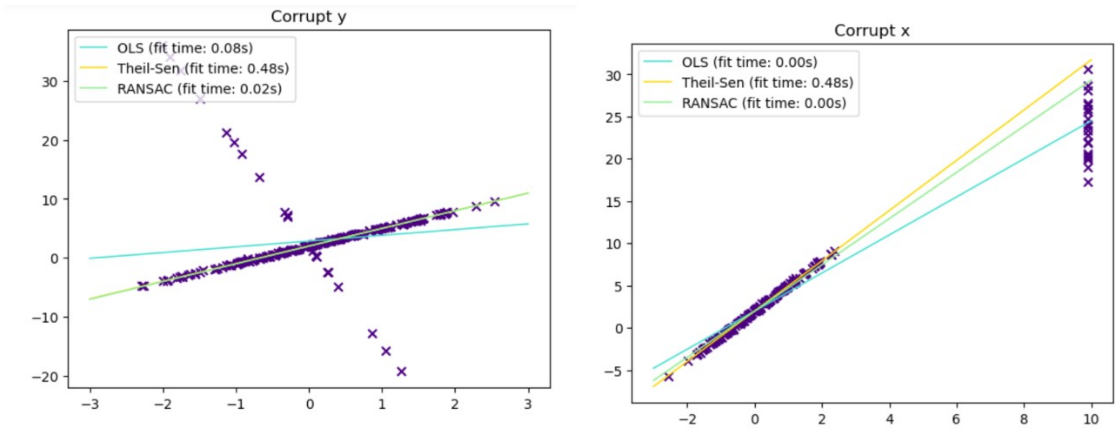

Рисунок 3 – Графики сравнения регрессий

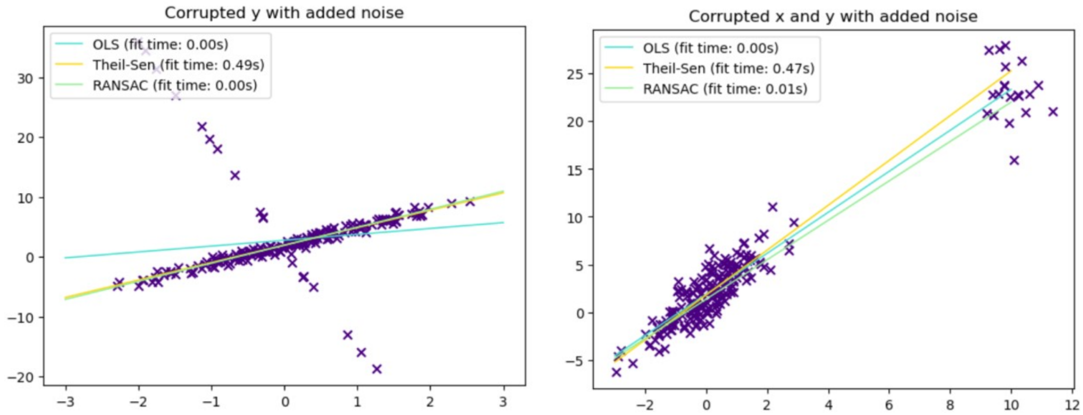

Рисунок 4 – Графики сравнения регрессий с добавлением дополнительного шума к данным

На основе анализа графиков можно сделать вывод о том, что регрессия Тейла-Сена более устойчива к выбросам, чем обычная линейная регрессия (OLS) и RANSAC-регрессия, потому что регрессия Тейла-Сена показывает более устойчивое поведение на графиках с "испорченными" данными.

На графике "Corrupt y" данные содержат выбросы в зависимой переменной (y), и регрессия Тейла-Сена показывает более стабильную и точную линию подгонки (fit line) по сравнению с обычной линейной регрессией (OLS) и RANSAC-регрессией. Это видно из того, что линия подгонки Тейла-Сена проходит ближе к "исправленным" данным, минуя выбросы.

На графике "Corrupt x" данные содержат выбросы в независимой переменной (x), и здесь также видно, что регрессия Тейла-Сена демонстрирует более устойчивую линию подгонки по сравнению с другими методами.

Аналогичные выводы можно сделать и по анализу графиков после внесенных изменений.

Таким образом, на основе этих графиков можно сделать вывод о более высокой устойчивости регрессии Тейла-Сена к выбросам.

Заключение

Анализируя добавленные изменения, можно сделать следующие выводы:

1. Регрессия Тейла-Сена демонстрирует относительно хорошее предсказание данных с учетом добавленного шума. Линия, соответствующая регрессии Тейла-Сена (золотистого цвета), достаточно близка к истинной зависимости данных (график "Corrupted y with added noise").

2. Регрессия Тейла-Сена демонстрирует хорошее предсказание данных, как для искаженных значений x, так и для y с добавленным шумом. Линия регрессии Тейла-Сена (золотистого цвета) снова близка к истинной зависимости данных (график "Corrupted x and y with added noise").

3. Добавление дополнительного шума может помочь создать более реалистичную модель, учитывая тот факт, что реальные данные часто содержат шумы и выбросы.

Таким образом, можно сделать вывод о том, что регрессия Тейла-Сена успешно справляется с предсказанием данных даже при наличии шумов и искажений в данных и добавление дополнительного шума к данным может помочь лучше понять поведение регрессионных моделей в условиях реальных данных и их устойчивость к шумам.

Список использованной литературы

1. Шестопал, О. В. Робастные методы получения адекватных статистических моделей// Известия высших учебных заведений. Северо-Кавказский регион. Технические науки. – ЮФУ (Ростов-на-Дону) ISSN: 0321-2653; DOI: 10.17213/0321-2653-2018-1. – 2018. – № 1(197). – С. 18-23.

2. Robert G. Staudte: Robust estimation and testing. Wiley, New York 1990. ISBN 0-471-85547-2

3. Барсегян, А. А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP: учебное пособие по специальности 071900 «Информационные системы и технологии» направления 654700 «Информационные системы» / А. А. Барсегян и др.; [гл. ред. Е. Кондукова] – СПб.: БХВ-Петербург, 2007. − 384 с.

4. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. Кн. 1. – М.: Финансы и статистика, 1983.

5. Кацко И.А., Паклин Н.Б. Практикум по анализу данных на М.: Колос, 2009, 278 с. : ил.–компьютере /Под ред. Е.В. Гореловой. (Учебники и учеб. Пособия для студентов высш. учеб. заведений).

6. L. Massaron, A. Boschetti, Regression Analysis with Python, Birmingham: Packt Publishing, 2016. – 312 с.