|  |  |  |  |
| --- | --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | | |
| Факультет | Компьютерных сетей и систем | | |
| Кафедра | Информатики  Дисциплина: Конструирование те технологии электронных вычислительных средств | | |
|  |  | | |
| ЛАБОРАТОРНАЯ РАБОТА №8  «Выявление аномалий» | | | |
| Магистрант:  гр. 858341  Жишкевич С.А. | |  | Проверил:  Стержанов М. В. |
| Минск, 2019 | | | |

# 

ХОД РАБОТЫ

**Задание.**

Набор данных ex8data1.mat представляет собой файл формата \*.mat (т.е. сохраненного из Matlab). Набор содержит две переменные X1 и X2 - задержка в мс и пропускная способность в мб/c серверов. Среди серверов необходимо выделить те, характеристики которых аномальные. Набор разделен на обучающую выборку (X), которая не содержит меток классов, а также валидационную (Xval, yval), на которой необходимо оценить качество алгоритма выявления аномалий. В метках классов 0 обозначает отсутствие аномалии, а 1, соответственно, ее наличие.

Набор данных ex8data2.mat представляет собой файл формата \*.mat (т.е. сохраненного из Matlab). Набор содержит 11-мерную переменную X - координаты точек, среди которых необходимо выделить аномальные. Набор разделен на обучающую выборку (X), которая не содержит меток классов, а также валидационную (Xval, yval), на которой необходимо оценить качество алгоритма выявления аномалий.

1. Загрузите данные ex8data1.mat из файла.
2. Постройте график загруженных данных в виде диаграммы рассеяния.
3. Представьте данные в виде двух независимых нормально распределенных случайных величин.
4. Оцените параметры распределений случайных величин.
5. Постройте график плотности распределения получившейся случайной величины в виде изолиний, совместив его с графиком из пункта 2.
6. Подберите значение порога для обнаружения аномалий на основе валидационной выборки. В качестве метрики используйте F1-меру.
7. Выделите аномальные наблюдения на графике из пункта 5 с учетом выбранного порогового значения.
8. Загрузите данные ex8data2.mat из файла.
9. Представьте данные в виде 11-мерной нормально распределенной случайной величины.
10. Оцените параметры распределения случайной величины.
11. Подберите значение порога для обнаружения аномалий на основе валидационной выборки. В качестве метрики используйте F1-меру.
12. Выделите аномальные наблюдения в обучающей выборке. Сколько их было обнаружено? Какой был подобран порог?
13. Ответы на вопросы представьте в виде отчета.

**Результат выполнения:**

1. Загрузите данные ex8data1.mat из файла.

1. data1 = scipy.io.loadmat('ex8data1.mat')
2. X1 = data1['X']
3. X1\_val = data1['Xval']
4. y1\_val = data1['yval']
5. **print**(X1.shape)

2. Постройте график загруженных данных в виде диаграммы рассеяния.



Рисунок 1 – график загруженных данных ex8data1.mat.

3. Представьте данные в виде двух независимых нормально распределенных случайных величин.



Рисунок 2 – гистограмма параметра «Задержка».



Рисунок 3 – гистограмма параметра «Пропускная способность».

4. Оцените параметры распределений случайных величин.

1. **def** get\_dist\_params(X):
2. **return** np.mean(X, axis=0), np.var(X, axis=0)
4. mu, sig2 = get\_dist\_params(X1)
6. **print**(f'\u03BC = {mu}')
7. **print**(f'\u03C3^2 = {sig2}')

5. Постройте график плотности распределения получившейся случайной величины в виде изолиний, совместив его с графиком из пункта 2.



Рисунок 4 – график плотности распределения получившейся случайной величины в виде изолиний.

6. Подберите значение порога для обнаружения аномалий на основе валидационной выборки. В качестве метрики используйте F1-меру.

Функция расчета меры f1:

1. **def** f1\_score(y\_true, y\_pred):
2. # print(f'y\_true = {y\_true.shape}, y\_pred = {y\_pred.shape}')
3. **assert** y\_true.shape == y\_pred.shape
5. tp = np.sum(np.logical\_and((y\_true == 1), (y\_pred == 1)))
6. fp = np.sum(np.logical\_and((y\_true == 0), (y\_pred == 1)))
7. fn = np.sum(np.logical\_and((y\_true == 1), (y\_pred == 0)))
9. **if** (tp + fp) == 0 **or** (tp + fn) == 0: **return** 0
11. precision = tp / (tp + fp)
12. recall = tp / (tp + fn)
14. result = 2 \* (precision \* recall) / (precision + recall)
16. **if** math.isnan(result):
17. result = 0
19. **return** result

Функция поиска наилучшего порога по мере f1:

1. **def** find\_eps(y\_true, p\_vals, iterations=100):
2. epsilons = np.linspace(np.max(p\_vals), np.min(p\_vals), iterations)
4. best\_f1 = 0
5. best\_eps = 1
6. best\_iteration = 0
8. **for** i **in** range(len(epsilons)):
9. eps = epsilons[i]
10. y\_pred = p\_vals < eps
11. f1 = float(f1\_score(y\_true, y\_pred))
12. **if** f1 > best\_f1:
13. best\_f1 = f1
14. best\_eps = eps
15. best\_iteration = i
17. **return** best\_eps, best\_f1, best\_iteration
18. p\_vals = p(X1\_val, mu, sig2)
19. best\_eps, best\_f1, best\_i = find\_eps(y1\_val, p\_vals, 200)
21. **print**(f'best\_eps: {best\_eps}')
22. **print**(f'best\_f1: {best\_f1}')
23. **print**(f'best\_iteration: {best\_i}')

Наилучшее значение «f1» получилось найти за 198 итераций. Мера f1 получилась 0.875, «эпсилон» - 0.000451.

7. Выделите аномальные наблюдения на графике из пункта 5 с учетом выбранного порогового значения.



Рисунок 5 – аномалии на графике плотности распределения.

8. Загрузите данные ex8data2.mat из файла.



9. Представьте данные в виде 11-мерной нормально распределенной случайной величины.



Рисунок 6 – гистограмма 11-мерной нормально распределенной случайной величины.

10. Оцените параметры распределения случайной величины.

Функция оценки использующая многомерное нормальное распределения:

1. **def** get\_dist\_params\_multi(X):
2. mu = np.mean(X, axis=0)
3. sig\_p = X - mu
4. Sigma = np.dot(sig\_p.T, sig\_p) / len(X)
6. **return** mu, Sigma



Рисунок 7 – параметры распределение 11-мерной случайной величины.

11. Подберите значение порога для обнаружения аномалий на основе валидационной выборки. В качестве метрики используйте F1-меру.

Функция поиска аномалий использующая многомерное распределеине.

1. **def** p\_multi(X, mu, Sigma):
2. m, n = X.shape
4. Sigma\_det = np.linalg.det(Sigma)
5. Sigma\_inv = np.linalg.pinv(Sigma)
7. e1 = 1 / (np.power((2 \* math.pi), n / 2) \* np.sqrt(Sigma\_det))
9. X\_mu = X - mu
11. e2 = np.exp(- 0.5 \* np.sum((np.dot(X\_mu, Sigma\_inv) \* X\_mu), axis=1))
13. **return** (e1 \* e2).reshape(-1, 1)

Подбор наилучшего значения порога:

1. p\_vals\_2 = p\_multi(X2\_val, mu\_2, Sigma\_2)
2. best\_eps\_2, best\_f1\_2, best\_i\_2 = find\_eps(y2\_val, p\_vals\_2, 6000)
4. **print**(f'best\_eps: {best\_eps\_2}')
5. **print**(f'best\_f1: {best\_f1\_2}')
6. **print**(f'best\_iteration: {best\_i\_2}')

Наилучшее значение «f1» получилось найти за 5998 итераций. Мера f1 получилась 0.736, «эпсилон» - 2.93e-19.

12. Выделите аномальные наблюдения в обучающей выборке. Сколько их было обнаружено? Какой был подобран порог?

При пороге «эпсилон» = 2.93e-19, на обучающей выборке было обнаружено 48 аномалий.

**Вывод.**

В ходе выполнения лабораторной работы я ознакомился задачей поиска аномалий в неразмеченных, и размеченных данных, а так же изучил алгоритмы которые помогают решить эту задачу. Для решения задач, поставленных в лабораторной работе, мной были реализованы алгоритмы поиска аномалий «с помощью распределения Гаусса» и «с помощью многомерного распределения Гаусса».