Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Отчёт по лабораторной работе №1

Метод парзеновского окна с фиксированным размером окна

Выполнила:

студент гр. ИП-813
Бурдуковский И.А.

Проверила:

Морозова К.И.

Новосибирск 2021

Оглавление

Задание

Входные данные:

К заданию на лабораторную работу прилагаются файлы, в которых представлены наборы данных из $\sim 10^4$ объектов. Каждый объект описывается двумя признаками ($f_j(x) \in R$) и соответствующим ему классом ($y \in \{0,1\}$).

Суть лабораторной работы заключается в написании классификатора на основе метода k ближайших соседей. Данные из файла необходимо разбить на две выборки, обучающую и тестовую, согласно общепринятым правилам разбиения. На основе этих данных необходимо обучить разработанный классификатор и протестировать его на обеих выборках. В качестве отчёта требуется представить работающую программу и таблицу с результатами тестирования для каждого из 10 разбиений. Разбиение выборки необходимо выполнять программно, случайным образом, при этом, не нарушая информативности обучающей выборки. Разбивать рекомендуется по следующему правилу: делим выборку на 3 равных части, 2 части используем в качестве обучающей, одну в качестве тестовой. Кроме того, обучающая выборка должна быть сгенерирована таким образом, чтобы минимизировать разницу между количеством представленных в ней объектов разных классов, т.е. $abs(|\{(x_i, y_i) \in X^l | y_i = -1\}| - |\{(x_i, y_i) \in X^l | y_i = 1\}|) \to min$.

Функция ядра: $T - треугольное\ K(x) = (1 - r)[r \leq 1]$

Файл с данными - 2

Результаты

Ход вычисления H:

```
Найдём оптимальное h
Для 2 процент точности = 95.78651685393258 %
Для 3 процент точности = 93.81701215153682 %
Для 4 процент точности = 92.76232616940582 %
Для 5 процент точности = 90.88569265707798 %
Для 6 процент точности = 90.10494752623688 %
Для 7 процент точности = 89.60883845924157 %
Для 8 процент точности = 89.5491191400418 %
Для 9 процент точности = 89.14925373134328 %
Для 10 процент точности = 89.04477611940298 %
```

Самый оптимальный результат получается при ширине окна равном двум. При таком размере окна на тестовой выборке вероятность правильной классификации равна 95,8%.

Код программы

```
import csv
from math import sqrt
from sklearn.model_selection import train_test_split


def loo(X_train, h_max: int = 10):
    h_calc = 0
    h_accuracy = 0

    for h in range(2, h_max + 1):
        correct = 0
        incorrect = 0

        for dot in X_train:
            class_0 = 0
            class_1 = 0

            for influence_dot in X_train:
                if not dot == influence_dot:
                    if core(distance(dot, influence_dot) / h) <= 1:
                        if influence_dot[2] == 0:
                            class_0 += 1
                        else:
                            class_1 += 1

            if class_0 + class_1 == 0:
                continue

            if class_0 >= class_1:
                calc = 0
            else:
                calc = 1

            # print(calc)

            if dot[2] == calc:
                correct += 1
            else:
                incorrect += 1

        accuracy = correct / (correct + incorrect)
        print("Для {0} процент точности = {1}".format(h, accuracy * 100), "%")
        if h_accuracy < accuracy:
            h_accuracy = accuracy
            h_calc = h
    return h_calc


def parzen(X_test, h):
    correct = 0

    for dot in X_test:
        class_0 = 0
        class_1 = 0
        for influence_dot in X_test:
            if not (dot[0] == influence_dot[0] and dot[1] == influence_dot[1]):
                if core(distance(dot, influence_dot) / h) <= 1:
                    if influence_dot[2] == 0:
```

```
                        class_0 += 1
                    else:
                        class_1 += 1
            if class_0 > class_1:
                calc = 0
            else:
                calc = 1

            if dot[2] == calc:
                correct += 1

        return correct


def distance(central_dot, influence_dot):
    return sqrt((central_dot[0] - influence_dot[0]) ** 2 + (central_dot[1] -
influence_dot[1]) ** 2)


def core(r):
    return abs(1 - r)


X_data = []
Y_data = []
with open("data2.csv") as f:
    csv_iter = csv.reader(f, delimiter=',')
    next(csv_iter)
    for row in csv_iter:
        X_data.append(row)

for row in X_data:
    row[0] = int(row[0])
    row[1] = int(row[1])
    row[2] = int(row[2])

X_train, X_test = train_test_split(X_data, test_size=0.33)

print("Найдём оптимальное h")
h = loo(X_train, 10)
print(h)
acc = parzen(X_test, h)
print("Точность на тестовой выборке {0}".format(acc / len(X_test) * 100), "%")
```