

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра программного обеспечения информационных технологий
Дисциплина: Методы и алгоритмы принятия решений (МиАПР)

ОТЧЕТ
по лабораторной работе №4

по теме:
«Классификация объектов на N классов методом персептрона»

Выполнил
студент: гр. 851006

Верещагин Н.В.

Проверил:

Марина И.М.

Минск 2021

СОДЕРЖАНИЕ

1	Постановка задачи.....	3
1.1	Цель работы	3
1.2	Исходные данные	3
1.3	Результат работы алгоритма	3
2	Алгоритм	4
3	Решение задачи.....	8

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Цель работы

Изучить особенности классификации объектов методом персептрона, а также научиться применять этот метод на практике.

1.2 Исходные данные

- N – Количество классов на которые требуется разделить объекты
- Обучающая выборка, представленная векторами с наборами признаков.

1.3 Результат работы алгоритма

N –решающих функций.

2 АЛГОРИТМ

После того, как получено N решающих функций, предъявляются объекты тестовой выборки, которые необходимо классифицировать, отнести к одному из классов. Тестовый объект подставляется в каждую из решающих функций и относится к тому из классов, где было получено максимальное значение.

Количество классов, обучающих объектов и их признаков может быть произвольным.

Допустим существование M решающих функций, характеризующихся тем свойством, что при $x \in \omega_i$, где x – объект, ω_i – класс $d_i(x) > d_j(x)$ для всех $i \neq j$.

Рассмотрим M классов $\omega_1, \omega_2, \dots, \omega_M$. Пусть на k -м шаге процедуры обучения системе предъявляется образ $x(k)$, принадлежащий классу ω_i . Вычисляются значения M решающих функций $d_j[x(k)] = w_j(k)x(k)$, $j = 1, 2, \dots, M$. Затем если выполняются условия $d_i[x(k)] > d_j[x(k)]$, $j = 1, 2, \dots, M; j \neq i$, то векторы весов не изменяются, т. е. $w_j(k+1) = w_j(k)$, $j = 1, 2, \dots, M$.

С другой стороны, допустим, что для некоторого l $d_l[x(k)] \leq d_i[x(k)]$. В этом случае выполняются следующие коррекции весов:

$$\begin{aligned}w_i(k+1) &= w_i(k) + cx(k), \\w_l(k+1) &= w_l(k) - cx(k), \\w_j(k+1) &= w_j(k), j = 1, 2, \dots, M; j \neq i, j \neq l,\end{aligned}\tag{1}$$

где c – положительная константа.

Если при рассмотрении случая 3 классы разделимы, то доказано, что этот алгоритм сходится за конечное число итераций при произвольных начальных векторах. Рассмотрим это на примере.

Даны классы, причем каждый из них содержит один образ: $\omega_1: \{(0, 0)\}$, $\omega_2: \{(1, 1)\}$, $\omega_3: \{(-1, 1)\}$. Дополним заданные образы: $(0, 0, 1)$, $(1, 1, 1)$, $(-1, 1, 1)$.

Выберем в качестве начальных векторов весов $w_1(1) = w_2(1) = w_3(1) = (0, 0, 0)$, положим $c = 1$ и, предъявляя образы в указанном порядке, получим следующее:

$$\begin{aligned}d_1[x(1)] &= w_1(1)x(1) = 0, \\d_2[x(1)] &= w_2(1)x(1) = 0, \\d_3[x(1)] &= w_3(1)x(1) = 0.\end{aligned}$$

Поскольку $x(1) \in \omega_1$ и $d_2[x(1)] = d_3[x(1)] = d_1[x(1)]$, первый весовой вектор увеличивается, а два других уменьшаются в соответствии с соотношениями (1), т. е.

$$w_1(2) = w_1(1) + x(1) = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$w_2(2) = w_2(1) - x(1) = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},$$

$$w_3(2) = w_3(1) - x(1) = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

Следующий предъявляемый образ $x(2) = (1, 1, 1)$ принадлежит классу ω_2 . Для него получаем

$$w_1(2)x(2) = 1, w_2(2)x(2) = -1, w_3(2)x(2) = -1.$$

Поскольку все произведения больше либо равны $w_2(2)x(2)$, вводятся корректировки векторов коэффициентов:

$$w_1(3) = w_1(2) - x(2) = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix},$$

$$w_2(3) = w_2(2) + x(2) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

$$w_3(3) = w_3(2) - x(2) = \begin{pmatrix} -1 \\ -1 \\ -2 \end{pmatrix}.$$

Следующий предъявленный образ $x(3) = (-1, 1, 1)$ принадлежит классу ω_3 . Для него получаем $w_1(3)x(3) = 0$, $w_2(3)x(3) = 0$, $w_3(3)x(3) = -2$. Все эти произведения опять требуют корректировки:

$$w_1(4) = w_1(3) - x(3) = \begin{pmatrix} 0 \\ -2 \\ -1 \end{pmatrix},$$

$$w_2(4) = w_2(3) - x(3) = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix},$$

$$w_3(4) = w_3(3) + x(3) = \begin{pmatrix} -2 \\ 0 \\ -1 \end{pmatrix}.$$

Поскольку в данном цикле итерации присутствовали ошибки, следует провести новый цикл. Положив $x(4) = x(1)$, $x(5) = x(2)$, $x(6) = x(3)$, получим $w_1(4)x(4) = -1$, $w_2(4)x(4) = -1$, $w_3(4)x(4) = -1$. Так как образ $x(4)$ принадлежит классу ω_1 , то все произведения «неверны». Поэтому

$$w_1(5) = w_1(4) + x(4) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(5) = w_2(4) - x(4) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(5) = w_3(4) - x(4) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Следующий предъявленный образ $x(5) = (1, 1, 1)$ принадлежит классу ω_2 . Соответствующие скалярные произведения равны $w_1(5)x(5) = -2$,

$w_2(5)x(5) = 0$, $w_3(5)x(5) = -4$. Образ $x(5)$ классифицирован правильно. Поэтому

$$w_1(6) = w_1(5) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(6) = w_2(5) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(6) = w_3(5) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Следующий образ $x(6)=(-1, 1, 1)$ принадлежит классу ω_3 , для него получаем $w_1(6)x(6)=-2$, $w_2(6)x(6)=-4$, $w_3(6)x(6)=-0$. Этот образ также классифицирован правильно, так что коррекции не нужны, т.е.

$$w_1(7) = w_1(6) = \begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix},$$

$$w_2(7) = w_2(6) = \begin{pmatrix} 2 \\ 0 \\ -2 \end{pmatrix},$$

$$w_3(7) = w_3(6) = \begin{pmatrix} -2 \\ 0 \\ -2 \end{pmatrix}.$$

Если продолжить процедуру обучения, рассматривая образы $x(7)$, $x(8)$, $x(9)$, можно убедиться, что в следующем полном цикле никакие коррекции не производятся. Поэтому искомые решающие функции имеют следующий вид:

$$d_1(x) = 0 \cdot x_1 - 2x_2 + 0 = -2x_2,$$

$$d_2(x) = 2x_1 - 0 \cdot x_2 - 2 = 2x_1 - 2,$$

$$d_3(x) = -2x_1 + 0 \cdot x_2 - 2 = -2x_1 - 2.$$

3 РЕШЕНИЕ ЗАДАЧИ

Число классов: 5

Число признаков объекта: 10

Число объектов в обучающей выборке для каждого класса: 2

Обучение

1 КЛАСС:
(-7, -2, 7, -8, 7, -1, -3, 0, -3, 3,)
(6, -2, 3, 3, 1, 9, -6, -3, -2, 9,)

2 КЛАСС:
(-1, 8, 2, -7, 6, 7, 3, 4, 3, 2,)
(6, -9, 5, -6, -8, -4, 5, 2, 4, -3,)

3 КЛАСС:
(-9, 5, -4, 0, -7, -4, -10, 6, -9, 4,)
(-8, -3, 9, -7, -6, -5, 5, 3, -4, 9,)

4 КЛАСС:
(6, 3, 7, 5, 6, 4, 3, 4, 1, -5,)
(-6, 2, 2, -10, 8, 3, 2, -9, -10, -1,)

5 КЛАСС:
(6, 7, 2, 8, -1, -1, -7, 2, -10, 6,)
(0, -8, -4, 4, 1, -3, 8, 0, 4, -1,)

РАЗДЕЛЯЮЩИЕ ФУНКЦИИ:
 $d(1) = 6x_1 - 18x_2 - 5x_3 + 15x_4 + 3x_5 + 4x_6 - 21x_7 - 19x_8 + 13x_9 - 1x_{10} - 7$
 $d(2) = 16x_1 + 6x_2 - 19x_3 - 15x_4 - 5x_5 + 8x_6 + 19x_7 + 7x_8 + 43x_9 - 23x_{10} - 5$
 $d(3) = -26x_1 - 5x_2 - 27x_3 + 11x_4 - 39x_5 - 30x_6 - 13x_7 + 5x_8 - 3x_9 + 1x_{10} - 7$
 $d(4) = 8x_1 + 0x_2 - 19x_3 + 15x_4 + 15x_5 + 2x_6 + 7x_7 - 23x_8 + 5x_9 - 37x_{10} - 7$
 $d(5) = 20x_1 - 12x_2 - 41x_3 + 49x_4 - 13x_5 - 20x_6 - 1x_7 - 9x_8 + 11x_9 - 15x_{10} - 7$

Классификация

4	-2	5	2	6	8	8	-4	5
---	----	---	---	---	---	---	----	---

Классифицировать

Данный вектор принадлежит 2 классу

Рисунок 1 – Пример работы программы 1

Число классов: 10

Число признаков объекта: 5

Число объектов в обучающей выборке для каждого класса: 3

Обучение

(-9, -4, -9, 5, 0,)
(1, -10, 4, 8, -6,)
7 КЛАСС:
(3, 3, -4, 8, 7,)
(6, -8, 2, 1, -5,)
(3, -4, -6, -6, 3,)
8 КЛАСС:
(4, 9, -7, -8, -5,)
(-8, 7, -2, 8, -7,)
(2, 9, -7, -8, 8,)
9 КЛАСС:
(6, 0, -9, 4, -8,)
(-9, -4, -9, 0, 0,)
(0, 1, 1, 4, -6,)
10 КЛАСС:
(5, -8, 7, -9, 5,)
(1, 7, 2, 2, -8,)
(5, -5, -3, 4, -1,)

РАЗДЕЛЯЮЩИЕ ФУНКЦИИ:
 $d(1) = -22123x_1 + 29511x_2 + 48753x_3 - 9687x_4 - 9432x_5 - 15908$
 $d(2) = -14719x_1 + 56277x_2 + 40081x_3 - 31819x_4 + 16485x_5 - 20030$
 $d(3) = -6311x_1 + 40927x_2 - 23467x_3 - 20015x_4 + 8483x_5 - 17650$
 $d(4) = -14283x_1 + 49591x_2 + 36893x_3 + 10133x_4 + 8587x_5 - 15928$
 $d(5) = -8675x_1 + 49515x_2 + 70641x_3 - 17499x_4 - 567x_5 - 19798$
 $d(6) = -24921x_1 + 17823x_2 + 33955x_3 - 14577x_4 + 8567x_5 - 16486$
 $d(7) = 5877x_1 + 39511x_2 + 36753x_3 - 11687x_4 + 14567x_5 - 15908$
 $d(8) = -30939x_1 + 77517x_2 + 41705x_3 - 16839x_4 + 4873x_5 - 15394$
 $d(9) = -30747x_1 + 50673x_2 + 27851x_3 - 9455x_4 - 9573x_5 - 17850$
 $d(10) = 1857x_1 + 48743x_2 + 61925x_3 - 20051x_4 - 5453x_5 - 16312$

Классификация

5	2	-6	3	8
---	---	----	---	---

Классифицировать

Данный вектор принадлежит 7 классу

Рисунок 2 – Пример работы программы 2

Число классов: 6

Число признаков объекта: 3

Число объектов в обучающей выборке для каждого класса: 2

Обучение

1 КЛАСС:
(-2, 1, 7,)
(-4, 4, -10,)

2 КЛАСС:
(5, -6, -3,)
(3, 7, -6,)

3 КЛАСС:
(-9, -5, 3,)
(1, -1, -5,)

4 КЛАСС:
(4, 2, -1,)
(1, 0, 3,)

5 КЛАСС:
(-2, -7, -9,)
(4, -4, -6,)

6 КЛАСС:
(0, -2, -9,)
(-1, 5, 1,)

РАЗДЕЛЯЮЩИЕ ФУНКЦИИ:
 $d(1) = -18308 \times x_1 + 6557 \times x_2 + 22587 \times x_3 - 5249$
 $d(2) = 7532 \times x_1 - 3643 \times x_2 + 15691 \times x_3 - 5333$
 $d(3) = -10256 \times x_1 - 7979 \times x_2 + 22875 \times x_3 - 6305$
 $d(4) = 964 \times x_1 + 921 \times x_2 + 29135 \times x_3 - 6885$
 $d(5) = 962 \times x_1 - 11093 \times x_2 + 19117 \times x_3 - 6883$
 $d(6) = -9036 \times x_1 + 2921 \times x_2 + 15135 \times x_3 - 4885$

Классификация

►	-4	2	-6

Классифицировать

Данный вектор принадлежит 5 классу

Рисунок 3 – Пример работы программы 3

Число классов: 7

Число признаков объекта: 5

Число объектов в обучающей выборке для каждого класса: 4

Обучение

1 КЛАСС:
(6, -4, 4, -10, -5,)
(-3, 9, -10, -9, -8,)
(0, 5, -7, -3, -6,)
(-1, -9, 8, 1, -1,)

2 КЛАСС:
(-8, -10, 3, 8, -10,)
(3, -1, -1, 7, -2,)
(-10, 6, -9, -10, -7,)
(2, 8, -5, -2, 9,)

3 КЛАСС:
(-3, -5, 7, -2, 8,)
(2, 6, -7, 9, 8,)
(7, -7, 6, 0, 8,)
(5, 2, 6, 9, -3,)

4 КЛАСС:
(8, 0, -7, 3, 0,)
(4, -4, -7, -7, 2,)
(0, 5, -1, 1, 0,)
(7, -7, -5, 9, -2,)

5 КЛАСС:
(7, 5, -1, 9, 0,)
(6, -10, 0, -9, -3,)
(9, -8, -5, -10, -9,)
(6, 9, -7, 4, -5,)

6 КЛАСС:
(7, 3, -10, -3, -3,)
(-9, 3, -6, -2, -9,)
(2, -3, 7, 0, 0,)
(2, -9, -6, -7, -5,)

7 КЛАСС:
(4, -7, -9, 5, 2,)
(5, 6, -1, -1, -6,)

Классификация

►	-4	2	-6	2

Классифицировать

Данный вектор принадлежит 2 классу

Рисунок 4 – Пример работы программы 4

Код программы:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```

namespace Lab4
{
    public class Persiptron
    {
        private readonly int classCount;

        private readonly int vectorsSize;

        public bool warningFlag;

        public Persiptron(int classCount, int vectorsSize)
        {
            this.classCount = classCount;
            this.vectorsSize = vectorsSize;
        }

        public Function[] GetSepareteFunctions(Vector[][] teachingVectors)
        {
            var result = EmptyFunctions();
            warningFlag = false;
            bool nextIteration = true;
            int iterationNumber = 0;
            while (nextIteration && iterationNumber < 1000)
            {
                iterationNumber++;
                nextIteration = DoOneIteration(teachingVectors, result);
            }
            if (iterationNumber == 1000) warningFlag = true;
            return result;
        }

        private Function[] EmptyFunctions()
        {
            var result = new Function[classCount];
            for (int i = 0; i < classCount; i++)
            {
                result[i] = new Function(vectorsSize);
            }
            return result;
        }

        private bool DoOneIteration(Vector[][] teachingVectors, Function[] re-
sult)
        {
            bool nextIteration = false;
            if (teachingVectors.Length != classCount) throw new Exception();
            for (int classNumber = 0; classNumber < classCount; classNumber++)
            {

```

```

        for (int i = 0; i < teachingVectors[classNumber].Length; i++)
        {
            if (WorkWithVector(teachingVectors[classNumber][i], re-
sult, classNumber))
            {
                nextIteration =true;
            }
        }
    }
    return nextIteration;
}

private bool WorkWithVector(Vector currentVector, Function[] re-
sult, int vectorsClass)
{
    var maxClass = GetMaxVectorClass(result, currentVector);
    if (maxClass != vectorsClass)
    {
        Panish(currentVector, result, vectorsClass);
        return true;
    }
    return false;
}

private void Panish(Vector currentVector, Function[] result, int vec-
torsClass)
{
    for (int i = 0; i < classCount; i++)
    {
        if (i == vectorsClass)
        {
            result[i] += currentVector;
        }
        else
        {
            result[i] += -1*currentVector;
        }
    }
}

public int GetMaxVectorClass(Function[] result, Vector currentVector)
{
    int max = result[0].GetValue(currentVector);
    int maxClass = 0;
    int maxCount = 1;
    for (int j = 1; j < classCount; j++)
    {
        int currentValue = result[j].GetValue(currentVector);
        if (currentValue > max)
        {

```

```

        maxCount = 0;
        max = currentValue;
        maxClass = j;
    }
    if (currentValue == max)
    {
        maxCount++;
    }
}
return maxCount == 1 ? maxClass : -1;
}
}
}

```