НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Теорія розпізнавання образів та класифікації в системах штучного інтелекту

Лабораторна робота №4

Виконав:

Студент групи КН-20002Б

Кропивка Анатолій Анатолійович

Київ 2023

Тема: Дослідження методу Байєса та Байєсівської процедури у розпізнаванні образів.

Мета: Засвоєння базових знань щодо статистичних методів розпізнавання образів на основі Байєсівських процедур і метода Байєса. Розробка програмної системи, що реалізує розпізнавання образів на основі зазначеного методу. Дослідження розробленої системи. Отримання практичних навичок з розробки програмних систем розпізнавання образів.

Підготовка до роботи: Вивчити й уяснити тереотичні відомості щодо способів класифікації образів на основі методу Байєса.

Хід роботи:

1. Розробити програмний застосунок на мові програмування C++, що реалізує розпізнавання образів на основі зазначеного методу.
2. Дослідити функціонування розробленого програмного застосунку на прикладах.
3. За результатами досліджень скласти звіт з описом отриманих результатів та обґрунтованими висновками.

Завдання: Класифікувати довільні реальні образи (об’єкти) – банки, фірми, інвестиційні фонди, кредитні спілки тощо за кількома показниками. Наприклад, власний капітал, прибуток і тому подібне.

Опис програми

Програма дозволяє провести дослідження та оцінку ефективності методу Байєса та Байєсівської процедури у розпізнаванні образів. Ці методи заснованні на використанні теореми Байєса для визначення ймовірності належності образу до кожного класу та використання цих ймовірностей для класифікації нових образів.

Після обчислення ймовірностей, програма використовує Байєсівськку процедуру для класифікаії нових вхідних образів. Вона порівнює ймовірності належності образу до різних класів та вибирає клас з найвищою ймовірністю.

Основа програми будується на попередніх лабораторних роботах.

Код програми

```
#define _USE_MATH_DEFINES

#include <Windows.h>

#include <chrono>
#include <cmath>
#include <cstdlib>
#include <format>
#include <iostream>
#include <stdexcept>
#include <string>
#include <thread>
#include <vector>

using namespace std;

double get_double()
{
    double value {};
    string str {};

    try {
        getline(cin, str);

        size_t position {};
        value = stod(str, &position);

        if (position != str.size()) {
            throw runtime_error("There are characters after the number!");
        }
```

```
        if (value < 0) {
            throw runtime_error("The entered value cannot be negative");
        }
    } catch (const exception& e) {
        cout << format("Exception: {}. The return value is 0!", e.what()) << endl;

        value = 0;
    }

    return value;
}

class Animal {
private:
    double height;
    double weight;

public:
    Animal()
        : height {}
        , weight {}
    {
    }

    Animal(double height, double weight)
        : height(height)
        , weight(weight)
    {
    }

    const double get_height() const
    {
```

```
        return height;
    }

    const double get_weight() const
    {
        return weight;
    }

    Animal from()
    {
        cout << "Enter the data about an animal." << endl;

        cout << "Height: ";
        height = get_double();

        cout << "Weight: ";
        weight = get_double();

        cout << endl;
        return *this;
    }
};

class Animals {
private:
    const char* name;
    vector<Animal> group;
    double ave_hg;
    double ave_wg;

public:
    Animals(const char* name, vector<Animal> animals)
```

```
    : name(name)
    , group(animals)
    , ave_hg {}
    , ave_wg {}
{
    for (auto& animal : animals) {
        ave_hg += animal.get_height();
        ave_wg += animal.get_weight();
    }

    ave_hg /= animals.size();
    ave_wg /= animals.size();
}

const double get_ave_hg() const
{
    return ave_hg;
}

const double get_ave_wg() const
{
    return ave_wg;
}

const vector<Animal>& get_group() const
{
    return group;
}

void output()
{
    cout << format("Average value of {}", name) << endl;
```

```
        cout << format("Height: {}", ave_hg) << endl;
        cout << format("Weight: {}", ave_wg) << endl;
        cout << endl;
    }
};

class Aim {
private:
    vector<Animals*> animals;
    Animal& obj;
    size_t size;
    long double f;

public:
    Aim(Animals& first, Animals& second, Animal& obj)
        : animals({ &first, &second })
        , obj(obj)
        , size(first.get_group().size() + second.get_group().size())
        , f {}
    {
        for (auto& group : animals) {
            f += function(*group);
        }
    }

    const long double probability(Animals& animals) const
    {
        return static_cast<long double>(animals.get_group().size()) / size;
    }

    const long double obj_minus_average(Animals& animals, const Animal& obj) const
    {
```

```
    double x { pow(obj.get_weight() - animals.get_ave_wg(), 2) };
    double y { pow(obj.get_height() - animals.get_ave_hg(), 2) };
    return y + x;
}

const long double dispersion(Animals& animals) const
{
    double value {};
    const vector<Animal>& ref_group { animals.get_group() };

    for (auto& animal : ref_group) {
        value += obj_minus_average(animals, animal);
    }

    value = value / ref_group.size();
    return value <= 0 ? 1.0 : value;
}

const long double fi(Animals& animals) const
{
    long double disper { dispersion(animals) };
    long double x { pow(2 * M_PI * disper, size / 2) };
    long double y { exp(-obj_minus_average(animals, obj) / (2 * disper)) };

    return 1.0 / x * y;
}

const long double function(Animals& animals) const
{
    return fi(animals) * probability(animals);
}
```

```
    const long double bayes(Animals& animals) const
    {
        return (probability(animals) * fi(animals)) / f;
    }

    const bool compare_bayes() const
    {
        const long double value_a { bayes(*animals[0]) };
        const long double value_b { bayes(*animals[1]) };

        cout << format("1: Bayes = {}", value_a) << endl;
        cout << format("2: Bayes = {}", value_b) << endl;

        return value_a > value_b ? true : false;
    }

    const long double get_f() const
    {
        return f;
    }
};

int main()
{
    thread go_away([]() {
        while (true) {
            this_thread::sleep_for(chrono::milliseconds(100));

            if (GetAsyncKeyState(VK_ESCAPE)) {
                exit(0);
            }
        }
```

```
});

Animals cats("cats",
    vector<Animal> {
        Animal(20, 4),
        Animal(24, 4.2),
        Animal(22, 4.6),
        Animal(31, 5),
        Animal(27, 5.1),
        Animal(29, 4.4),
    });

Animals dogs("dogs",
    vector<Animal> {
        Animal(40, 6),
        Animal(60, 12),
        Animal(100, 20),
        Animal(55, 9),
        Animal(69, 15),
        Animal(36, 7),
    });

cats.output();
dogs.output();

while (true) {
    cout << endl;
    cout << "The data of the test object." << endl;

    auto obj { Animal().from() };
    auto aim { Aim(cats, dogs, obj) };
```

```
        cout << format("1: Dispersion = {}", aim.dispersion(cats)) << endl;
        cout << format("1: p = {}", aim.probability(cats)) << endl;
        cout << format("1: (X - C)^2 = {}", aim.obj_minus_average(cats, obj)) << endl;
        cout << format("1: fi = {}", aim.fi(cats)) << endl;
        cout << endl;
        cout << format("2: Dispersion = {}", aim.dispersion(dogs)) << endl;
        cout << format("2: p = {}", aim.probability(dogs)) << endl;
        cout << format("2: (X - C)^2 = {}", aim.obj_minus_average(dogs, obj)) << endl;
        cout << format("2: fi = {}", aim.fi(dogs)) << endl;
        cout << endl;
        cout << format("F = {}", aim.get_f()) << endl;

        if (aim.compare_bayes()) {
            cout << "The test object is cat" << endl;
        } else {
            cout << "The test object is dog" << endl;
        }
    }

    return 0;
}
```

Знімки екрану

```
PS D:[redacted]\TheoryOfRecognizeImages\MyLabs\lab_4> ./bin/main.exe
Average value of cats
Height: 25.5
Weight: 4.55

Average value of dogs
Height: 60
Weight: 11.5


The data of the test object.
Enter the data about an animal.
Height: 26.6
Weight: 12.2

1: Dispersion = 15.0758333333333300885
1: p = 0.5
1: (X - C)^2 = 59.7324999999999946
1: fi = 1.90926386097078357707e-13

2: Dispersion = 470.58333333333331439
2: p = 0.5
2: (X - C)^2 = 1116.0499999999999545
2: fi = 4.572023156903952819e-22

F = 9.546319327714033638e-14
1: Bayes = 0.99999999760534767385
2: Bayes = 2.39465232617499908886e-09
The test object is cat
```

```
The data of the test object.
Enter the data about an animal.
Height: 99.9
Weight: 55.5

1: Dispersion = 15.0758333333333300885
1: p = 0.5
1: (X - C)^2 = 8131.2625000000007276
1: fi = 1.050166614040355634 6e-129

2: Dispersion = 470.58333333333331439
2: p = 0.5
2: (X - C)^2 = 3528.0100000000002183
2: fi = 3.5247401199358858384e-23

F = 1.76237005996794291 92e-23
1: Bayes = 2.979415725149852775e-107
2: Bayes = 1
The test object is dog
```

Висновок: Програма надає можливість дослідити та експериментувати з методом Байєса та Байєсівською процедурою у розпізнаванні образів. Вона сприяє кращому розумінню теорії розпізнавання образів та надає можливість провести практичне дослідження методу Байєса та Байєсівської процедури у реальних задачах класифікацій образів.