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 №1

по теме:

«Распознавание образов на основе контролируемого обучения»

Выполнил

студент: гр. 851006 Верещагин Н.В.

Проверил: Марина И.М.

Минск 2021

Содержание

[1 Постановка задачи 3](#_Toc63794150)

[1.1 Цель работы 3](#_Toc63794151)

[1.2 Исходные данные 3](#_Toc63794152)

[1.3 Цели и результат работы алгоритма 3](#_Toc63794153)

[2 Алгоритм К-средних 4](#_Toc63794154)

[3 Решение задачи 5](#_Toc63794155)

# Постановка задачи

## Цель работы

Изучить особенности методов распознавания образов, использующих контролируемое обучение, и научиться классифицировать объекты с помощью алгоритма *К-средних.*

## Исходные данные

* Количество образов в диапазоне от 1000 до 100 000.
* Число классов – от 2 до 20.
* Признаки объектов задаются случайным образом, это координаты векторов.
* *К* элементов из набора векторов случайным образом назначают центрами классов.

## Цели и результат работы алгоритма

Определить ядрами классов *К* типичных представителей классов и максимально компактно распределить вокруг них остальные объекты выборки.

# Алгоритм К-средних

1. Фиксируется *К* ядер (центров областей). Вокруг формируются области по правилу минимального расстояния. На *r*-м этапе вектор связывается с ядром, если удовлетворяется неравенство:

2. На (r+1)-м этапе определяются новые элементы, характеризующие новые ядра За их значения принимают векторы , обеспечивающие минимум среднеквадратичного отклонения:

*Ji* принимает минимальное значение лишь на одном , равном среднему арифметическому векторов, принадлежащих одной области .

3. Если хотя бы в одной из областей поменялось положение ядра, то пересчитываются области принадлежащих им векторов, т.е. определяются расстояния от объектов (не ядер) до новых ядер. В результате этого может произойти перераспределение областей. Затем повторяется шаг 2. Процедура заканчивается, если на (r+1)-м шаге ее выполнения положения центров областей не меняются по сравнению с r-м шагом.

# Решение задачи

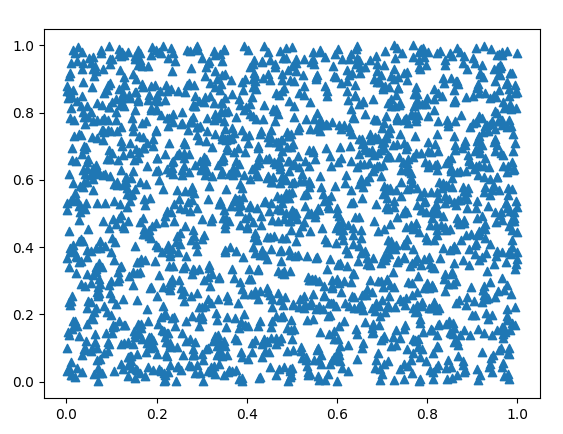


Рисунок 1 – 1 итерация

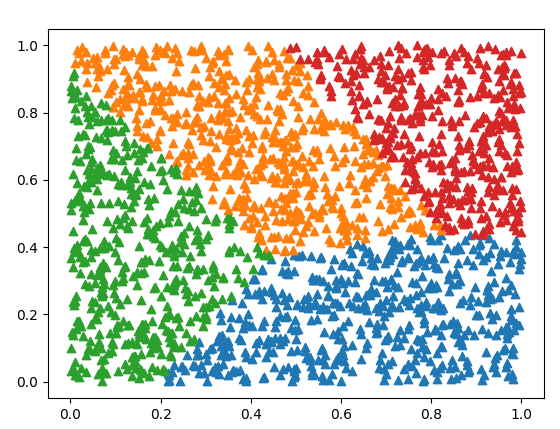


Рисунок 2 – 2 итерация

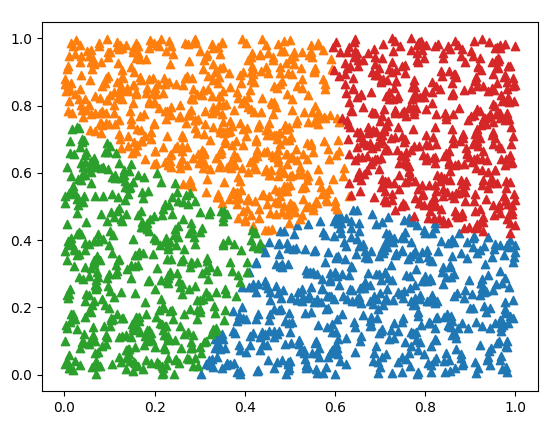


Рисунок 3 – 3 итерация

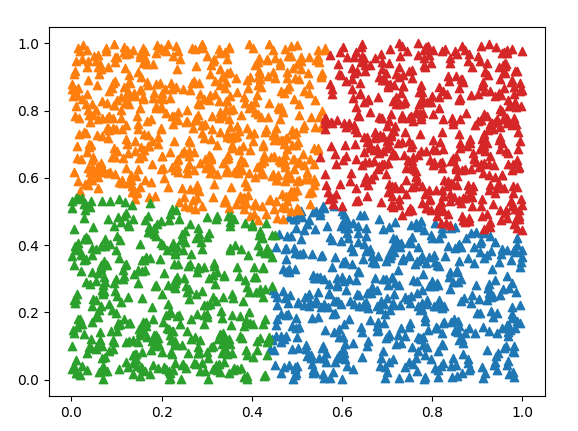


Рисунок 4 – 4 итерация

**Код программы:**

*import* matplotlib.pyplot *as* plt

*import* numpy

DIM = 2

N = 2000

num\_cluster = 10

condition = 10

x = numpy.random.rand(N, DIM)

y = numpy.zeros(N)

*for* t *in* range(condition):

    input()

*for* k *in* range(num\_cluster):

        fig = plt.scatter(x[y == k, 0], x[y == k, 1], marker='^')

    plt.show()

*if* t == 0:

        index\_ = numpy.random.choice(range(N), num\_cluster, replace=False)

        mean = x[index\_]

*else*:

*for* k *in* range(num\_cluster):

            mean[k] = numpy.mean(x[y == k], axis=0)

*for* i *in* range(N):

        dist = numpy.sum((mean - x[i])\*\*2, axis=1)

        pred = numpy.argmin(dist)

        y[i] = pred