НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Дисциплина: «Анализ данных»

Домашнее задание на тему:
«Лабораторная работа №2»

Выполнил: Осипов Лев,
студент группы 301ПИ (1).

Москва, 2015 г.

СОДЕРЖАНИЕ

Теоретическая часть **3**

Задание 1 ..... **3**

Задание 2 ..... **3**

Задание 3 ..... **3**

Практическая часть **3**

Задание 1 ..... **3**

Задание 2 ..... **4**

Задание 3 ..... **6**

Список литературы **8**

Текст программы **9**

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

ЗАДАНИЕ 1

Для применения метода ближайшего соседа к данным с номинальными признаками можно представить каждый признак как значение из множества {0,1} или же {0,n}, если используем взвешенные признаки. С таким подходом каждый признак будет представлять собой измерение.

ЗАДАНИЕ 2

Да, проблема возникнет. Дело в том, что будут считаться равнозначными равные по номиналу значения пульса и веса, которые измеряются соответственно в ударах в минуту и граммах. Для того чтобы этого избежать, можно привести все к одной единице измерения с помощью коэффициента, который можно получить, например, поделив длины интервалов значений признаков.

ЗАДАНИЕ 3

Сложность будет O(ns), так как расстояния для нулевых признаков считать не нужно (расстояние будет квадратом входного значения). Итерирование происходит только по s признакам каждого объекта.

ПРАКТИЧЕСКАЯ ЧАСТЬ

ЗАДАНИЕ 1

Для решения задания были посчитаны средние максимальные и минимальные значений косинусов углов между случайно сгенерированных точками разных размерностей.

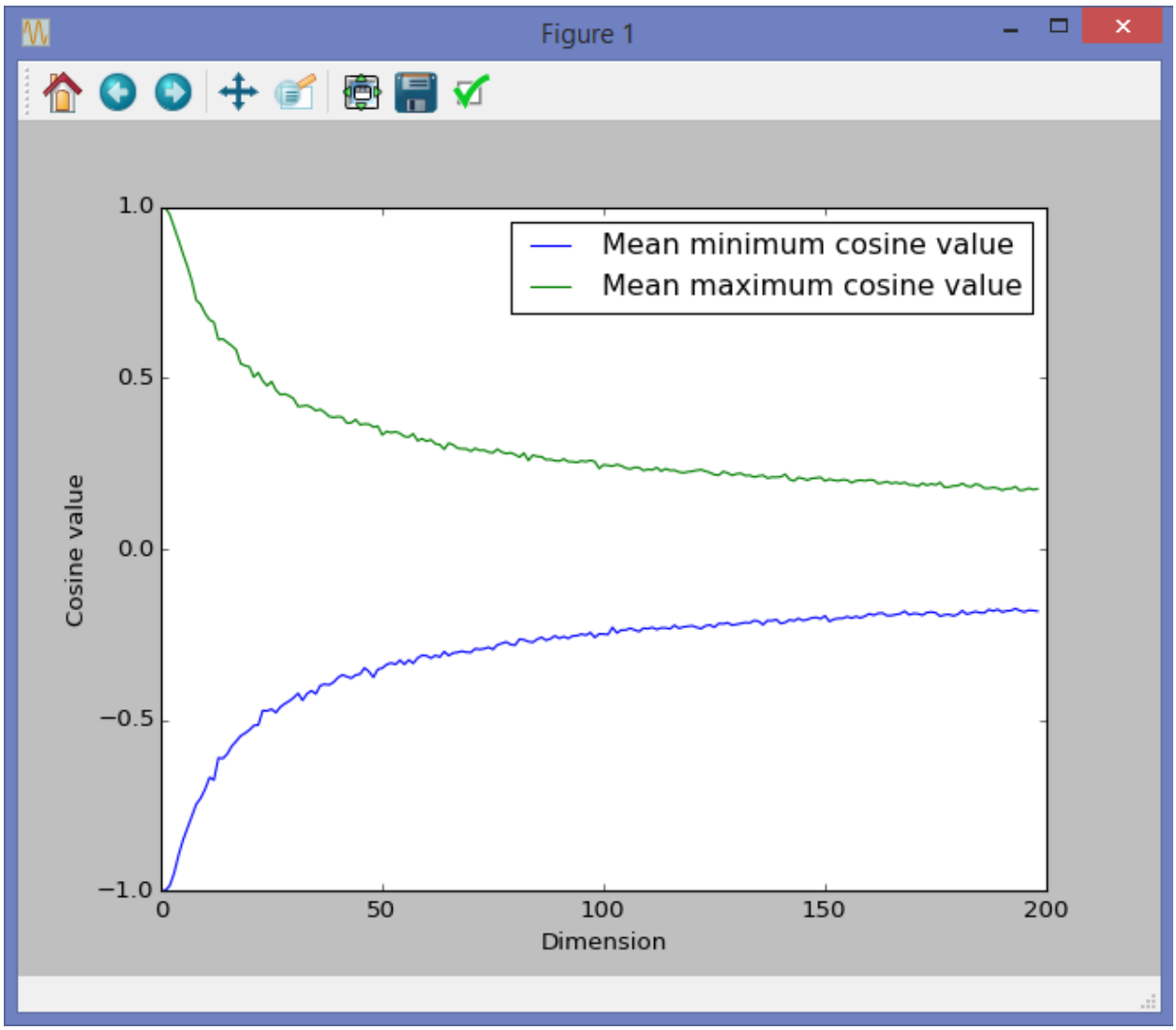

Рис. 1

По результатам программы (Рис. 1) видно, что средние с увеличением размерности стремятся к нулю, что как раз и говорит о том, что все точки становятся «практически ортогональны».

ЗАДАНИЕ 2

Для решения задания были посчитаны значения l_1, l_2, l_{inf} для случайно сгенерированных точек разных размерностей и для каждого из значений был произведен расчет по следующей формуле: $\frac{\max(norm(x)) - \min(norm(x))}{\max(norm(x))}$

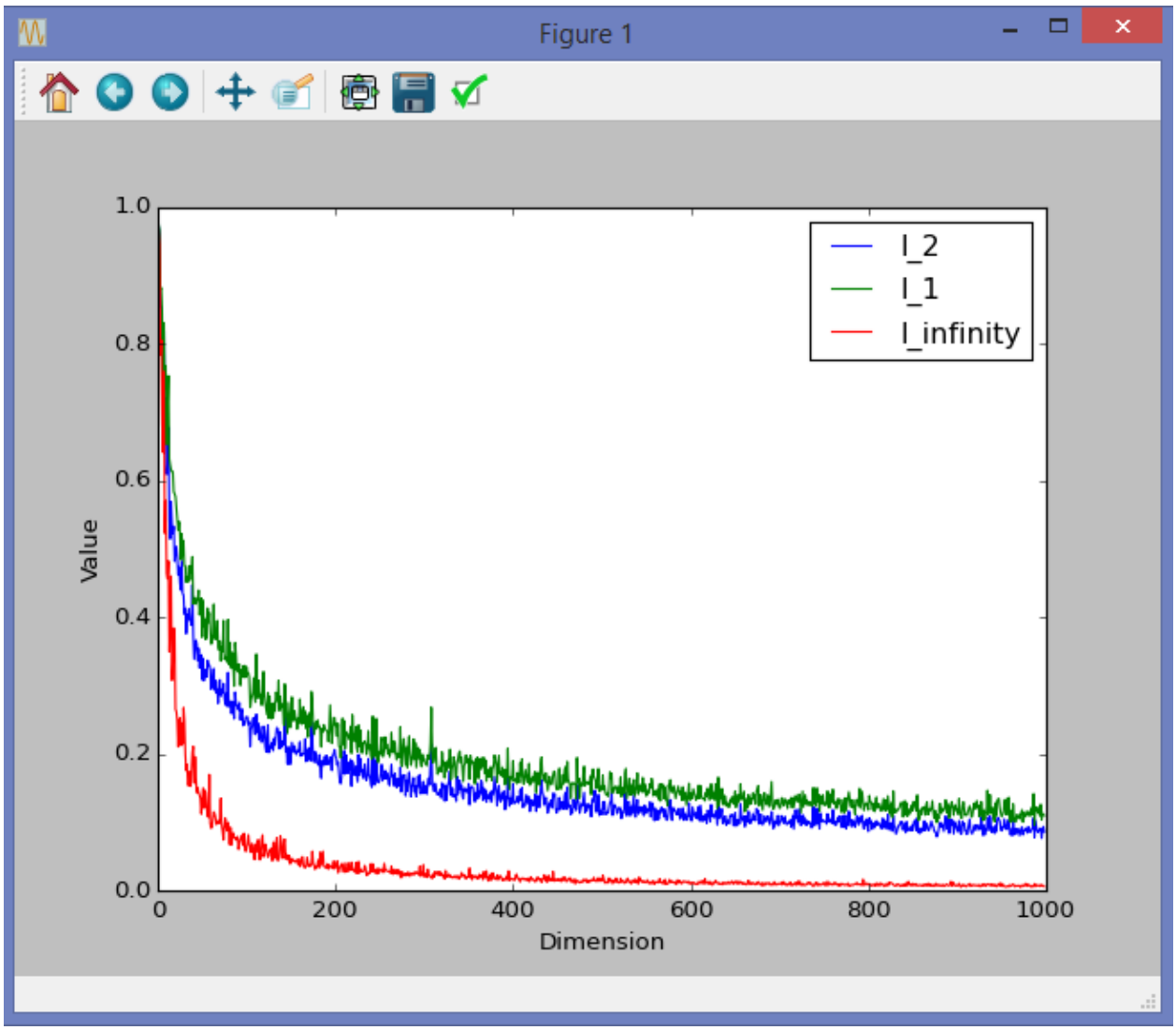

Рис. 2

Если считать, что каждое из значений признаков независимо относительно выборки координат определенной точки и все значения имеют одинаковое распределение, то можно увидеть тенденцию стремления значений норм к одному значению с ростом размерности пространства. Именно поэтому величина, полученная при расчете, с увеличением размерности стремится к нулю, что можно наблюдать на графике, полученным с помощью написанной программы (Рис. 2).

Из вышесказанного можно сделать вывод, что с увеличением размерности пространства расстояние между точками становится неинформативным.

ЗАДАНИЕ 3

Для решения задания была написана программа поиска похожих ресторанов, основанная на методе ближайшего соседа.

Примеры работы программы представлены на Рис. 3, Рис. 4 и Рис. 5.

```
Restaurant:  Sazerac House ['American (Regional)\n' 'Cajun\n' 'Creole\n' 'Good Decor\n'
 'Dining Outdoors\n' 'Excellent Food\n' '$15-$30\n' 'Excellent Service\n'
 'Warm spots by the fire\n']
Similar:
Benito I ['Good Decor\n' 'Excellent Food\n' '$15-$30\n' 'Excellent Service\n']
Pietro & Vanessa ['Good Decor\n' 'Dining Outdoors\n' 'Excellent Food\n' 'Italian\n'
 '$15-$30\n' 'Excellent Service\n']
Paolucci's ['Good Decor\n' 'Excellent Food\n' 'Italian\n' '$15-$30\n'
 'Excellent Service\n' 'Warm spots by the fire\n']
Swing Street Cafe ['Good Decor\n' 'Excellent Food\n' '$15-$30\n' 'Excellent Service\n']
Joe & Joe ['Good Decor\n' 'Excellent Food\n' 'Italian\n' '$15-$30\n'
 'Excellent Service\n']
Volare ['Good Decor\n' 'Excellent Food\n' 'Italian\n' '$15-$30\n'
 'Excellent Service\n']
Pasticcio ['Good Decor\n' 'Excellent Food\n' 'Italian\n' '$15-$30\n'
 'Excellent Service\n']
Four Winds ['Good Decor\n' 'Excellent Food\n' 'Japanese\n' '$15-$30\n'
 'Excellent Service\n']
Pier 25A ['Good Decor\n' 'Excellent Food\n' '$15-$30\n' 'Excellent Service\n'
 'Seafood\n']
Crepes Suzette ['Good Decor\n' 'Excellent Food\n' 'French Bistro\n' '$15-$30\n'
 'Excellent Service\n']
```

Рис. 3. Результат работы программs на ресторане «Sazerac House».

```
Restaurant:  Isabella's ['After Hours Dining\n' 'Excellent Decor\n' 'Excellent Food\n' 'Italian\n'
 'Late Night Menu\n' 'Mediterranean\n' '$15-$30\n' 'Place for Singles\n'
 'Good Service\n' 'Special Brunch Menu\n' 'Weekend Dining\n']
Similar:
Quartiere ['After Hours Dining\n' 'Excellent Decor\n' 'Excellent Food\n' 'Italian\n'
 'Late Night Menu\n' '$15-$30\n' 'Good Service\n']
Meridiana ['Excellent Decor\n' 'Excellent Food\n' 'Italian\n' '$15-$30\n'
 'Good Service\n']
Pellegrino ['After Hours Dining\n' 'Excellent Decor\n' 'Excellent Food\n' 'Italian\n'
 'Late Night Menu\n' 'Little Known But Well Liked\n' '$15-$30\n'
 'Excellent Service\n']
Mary's Restaurant ['After Hours Dining\n' 'Excellent Decor\n' 'Excellent Food\n' 'Italian\n'
 'Late Night Menu\n' '$15-$30\n' 'Excellent Service\n'
 'Warm spots by the fire\n']
La Jumelle ['After Hours Dining\n' 'Excellent Decor\n' 'Excellent Food\n'
 'French Bistro\n' 'Hip Place To Be\n' 'Late Night Menu\n'
 'Mediterranean\n' '$15-$30\n' 'Place for Singles\n' 'Prix Fixe Menus\n'
 'Good Service\n' 'Warm spots by the fire\n']
101 ['Excellent Decor\n' 'Excellent Food\n' 'Italian\n' '$15-$30\n'
 'Excellent Service\n']
Cucina Stagionale ['After Hours Dining\n' 'Good Decor\n' 'Excellent Food\n'
 'For the Young and Young at Heart\n' 'Italian\n' 'Late Night Menu\n'
 '$15-$30\n' 'Good Service\n' 'Weekend Dining\n']
Stellina ['Good Decor\n' 'Excellent Food\n' 'Italian\n' '$15-$30\n' 'Good Service\n']
103 NYC ['Good Decor\n' 'Excellent Food\n' '$15-$30\n' 'Place for Singles\n'
 'Good Service\n']
Vinsanto ['Excellent Decor\n' 'Excellent Food\n' 'Italian\n' '$15-$30\n'
 'Excellent Service\n']
```

Рис. 4. Результат работы программы на ресторане «Isabella's».

```
Restaurant:  Lucky Strike ['After Hours Dining\n' 'American (Traditional)\n' 'Good Decor\n'
 'Good Food\n' 'French Bistro\n' 'Late Night Menu\n' '$15-$30\n'
 'Place for Singles\n' 'Good Service\n']
Similar:
Koo Koo's Bistro ['After Hours Dining\n' 'American (Contemporary)\n' 'Good Decor\n'
 'Good Food\n' 'French Bistro\n' 'Late Night Menu\n' '$15-$30\n'
 'Good Service\n']
Florent ['After Hours Dining\n' 'Good Decor\n' 'Excellent Food\n' 'French Bistro\n'
 'Late Night Menu\n' '$15-$30\n' 'Good Service\n']
Roebling's ['American (Traditional)\n' 'Good Decor\n' 'Good Food\n' '$15-$30\n'
 'Good Service\n' 'Seafood\n']
Bill's Gay 90's ['Good Decor\n' 'Good Food\n' '$15-$30\n' 'Good Service\n']
Moran's ['American (Traditional)\n' 'Good Decor\n' 'Good Food\n' '$15-$30\n'
 'Good Service\n' 'Steakhouses\n' 'Warm spots by the fire\n']
103 NYC ['Good Decor\n' 'Excellent Food\n' '$15-$30\n' 'Place for Singles\n'
 'Good Service\n']
Il Vagabondo ['Good Decor\n' 'Good Food\n' 'Italian\n' '$15-$30\n' 'Good Service\n']
Lexington Avenue Grill ['Classic Hotel Dining\n' 'Good Decor\n' 'Good Food\n' '$15-$30\n'
 'Good Service\n']
Lucy's Retired Surfers ['After Hours Dining\n' 'Good Decor\n' 'Good Food\n' 'Late Night Menu\n'
 'Mexican\n' '$15-$30\n' 'Place for Singles\n' 'Good Service\n' 'Tex-Mex\n']
Sharkey's ['American (Traditional)\n' 'Good Decor\n' 'Good Food\n' '$15-$30\n'
 'Excellent Service\n']
```

Рис. 5. Результат работы программы на ресторане «Lucky Strike».

СПИСОК ЛИТЕРАТУРЫ

1) **Анализ данных (Программная инженерия)** – http://wiki.cs.hse.ru/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85_%28%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B8%D1%8F%29#.D0.9E.D1.84.D0.BE.D1.80.D0.BC.D0.BB.D0.B5.D0.BD.D0.B8.D0.B5_.D0.BF.D0.B8.D1.81.D0.B5.D0.BC

ТЕКСТ ПРОГРАММЫ

```
__author__ = 'Lev Osipov'

import numpy as np
import matplotlib.pyplot as plt
from heapq import nsmallest


def cosine(vector1, vector2):
    return np.dot(vector1, vector2) / (np.linalg.norm(vector1) *
np.linalg.norm(vector2))

# Task 1
points = 100
dimension = 200
coordinates = []
results = []
print "Wait, please.."
for dim in xrange(1, dimension):
    coordinates.append(np.random.rand(points, dim) * 2 - 1)
    dim_results = []
    for i in xrange(points):
        cos = []
        for j in xrange(points):
            if i != j:
                cos.append(cosine(coordinates[dim - 1][i],
coordinates[dim - 1][j]))
        dim_results.append([np.min(cos), np.max(cos)])
    results.append(np.mean(dim_results, axis=0))
np_results = np.array(results)
plt.plot(np_results[:, 0], label="Mean minimum cosine value")
plt.plot(np_results[:, 1], label="Mean maximum cosine value")
plt.xlabel('Dimension')
plt.ylabel('Cosine value')
plt.legend()
plt.show()

# Task 2
points = 1000
dimension = 1000
coordinates = []
function = []
print "Wait, please.."
for dim in xrange(1, dimension):
    coordinates.append(np.random.rand(points, dim) * 2 - 1)
    l2 = []
    l1 = []
    l_inf = []
    for i in xrange(points):
        l2.append(np.linalg.norm(coordinates[dim - 1][i]))
        l1.append(np.linalg.norm(coordinates[dim - 1][i], 1))
        l_inf.append(np.linalg.norm(coordinates[dim - 1][i],
```

```
np.inf))
    l2_max = np.max(l2)
    l1_max = np.max(l1)
    l_inf_max = np.max(l_inf)
    l2_min = np.min(l2)
    l1_min = np.min(l1)
    l_inf_min = np.min(l_inf)
    function.append([(l2_max - l2_min) / l2_max, (l1_max -
l1_min) / l1_max, (l_inf_max - l_inf_min) / l_inf_max])
np_function = np.array(function)
plt.plot(np_function[:, 0], label="l_2")
plt.plot(np_function[:, 1], label="l_1")
plt.plot(np_function[:, 2], label="l_infinity")
plt.xlabel('Dimension')
plt.ylabel('Value')
plt.legend()
plt.show()

# Task 3
features_file = open("features.txt")
feature_list = []
for feature in features_file:
    id, name = feature.split('\t')
    feature_list.append(name)
features_file.close()

restaurants_file = open("new_york.txt")
restaurants = {}  # dictionary
for restaurant in restaurants_file:
    id, name, features = restaurant.split('\t')
    current_features = np.zeros(len(feature_list))
    current_features[[int(current_feature) for current_feature
in features.split()]] = 1  # matching existing features
    restaurants[name] = current_features
restaurants_file.close()

fl = np.array(feature_list)


def find_similar(restaurant_name, all_restaurants,
features_array):
    cf = all_restaurants[restaurant_name]  # current features
    del all_restaurants[restaurant_name]  # deleting current
restaurant
    similar_ones = nsmallest(10, all_restaurants.items(), lambda
item: np.linalg.norm(item[1] - cf))  # finding
    print "Restaurant: ", restaurant_name,
features_array[np.nonzero(cf)]
    print "Similar:"
    for similar in similar_ones:
        print similar[0], features_array[np.nonzero(similar[1])]
```

```
# Tests
find_similar("Sazerac House", restaurants.copy(), fl)
find_similar("Isabella's", restaurants.copy(), fl)
find_similar("Lucky Strike", restaurants.copy(), fl)
```