|  |  |  |  |
| --- | --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | | |
| Факультет | Компьютерных сетей и систем | | |
| Кафедра | Информатики  Дисциплина: Конструирование те технологии электронных вычислительных средств | | |
|  |  | | |
| **лабораторная работа №9**  по курсу Машинное обучение  на тему  **Рекомендательные системы** | | | |
| Магистрант:  гр. 858341  Кудин Н.И. | |  | Проверил:  Стержанов М.В. |
| Минск, 2019 | | | |

**Постановка задачи**

Набор данных ex9\_movies.mat представляет собой файл формата \*.mat (т.е. сохраненного из Matlab). Набор содержит две матрицы Y и R - рейтинг 1682 фильмов среди 943 пользователей. Значение Rij может быть равно 0 или 1 в зависимости от того оценил ли пользователь j фильм i. Матрица Y содержит числа от 1 до 5 - оценки в баллах пользователей, выставленные фильмам.

**Ход работы**

1. Загрузим данные ex9\_movies.mat из файла.
2. Выберем число признаков фильмов (n) для реализации алгоритма коллаборативной фильтрации, n = 3.
3. Реализуем функцию стоимости для алгоритма.

import numpy as np  
  
def cofiCostFunc(params, Y, R, num\_users, num\_movies, num\_features, lambda\_var):  
  
 X = np.reshape(params[:num\_movies\*num\_features], (num\_movies, num\_features), order='F')  
 Theta = np.reshape(params[num\_movies\*num\_features:], (num\_users, num\_features), order='F')  
  
 # You need to return the following values correctly  
 J = 0  
 X\_grad = np.zeros(X.shape)  
 Theta\_grad = np.zeros(Theta.shape)  
  
  
 # X \* Theta performed according to low rank matrix vectorization  
 squared\_error = np.power(np.dot(X,Theta.T) - Y,2)  
  
 # for cost function, sum only i,j for which R(i,j)=1  
 J = (1/2.) \* np.sum(squared\_error \* R)  
  
  
  
 X\_grad = np.dot(( np.dot(X, Theta.T) - Y ) \* R, Theta)  
  
  
 # Theta\_grad is of dimensions n\_u x n, where n\_u = 943 and n = 100   
 # ( (X \* Theta') - Y ) is n\_m x n\_u   
 # (( (X \* Theta') - Y ) .\* R) is still n\_m x n\_u   
  
 Theta\_grad = np.dot((( np.dot(X, Theta.T) - Y ) \* R).T, X)  
  
  
 J = J + (lambda\_var/2.)\*( np.sum( np.power(Theta, 2) ) + np.sum( np.power(X, 2) ) )  
  
 X\_grad = X\_grad + lambda\_var\*X  
 Theta\_grad = Theta\_grad + lambda\_var\*Theta  
  
 grad = np.concatenate((X\_grad.reshape(X\_grad.size, order='F'), Theta\_grad.reshape(Theta\_grad.size, order='F')))  
  
 return J, grad

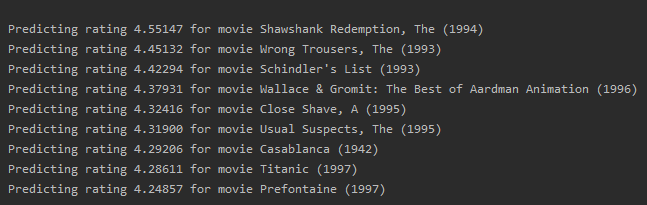
1. Реализуем функцию вычисления градиентов с использованием векторизации для ускорения процесса обучения. Установим L2-регуляризацию в модель со значением 10.

maxiter = 100  
options = {'disp': True, 'maxiter':maxiter}  
lambda\_var=10  
  
# Create "short hand" for the cost function to be minimized  
def costFunc(initial\_parameters):  
 return ccf.cofiCostFunc(initial\_parameters, Y, R, num\_users, num\_movies, num\_features, lambda\_var)

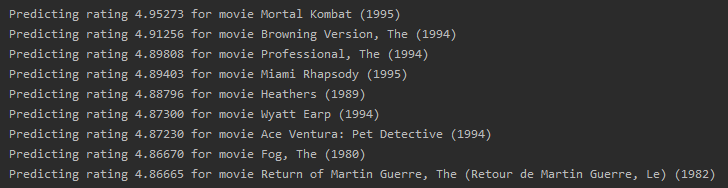
1. Обучим модель рекомендаций про помощи метода «L-BFGS-B».

results = minimize(costFunc, x0=initial\_parameters, options=options, method="L-BFGS-B", jac=True)  
theta = results["x"]  
  
# Unfold the returned theta back into U and W  
X = np.reshape(theta[:num\_movies\*num\_features], (num\_movies, num\_features), order='F')  
Theta = np.reshape(theta[num\_movies\*num\_features:], (num\_users, num\_features), order='F')

1. Добавим свои оценки 5 к фильмам: из файла **movie\_ids.txt,** который содержит индексы каждого из фильмов. Были выставлены следующие оценки (фильмы взяты из одного жанра для явности эксперимента):
   * Four Rooms (1995) – 5
   * Bad Boys (1995) – 5
2. Получим следующий результат по рекомендованным фильмам:



1. Обучим модель с помощью сингулярного разложения матриц. Получили следующий результат:



**Заключение**

Результаты рекомендаций двух подходов обучения модели отличаются между собой, однако прослеживается взаимосвязь жанров рекомендованных фильмов с оцененными для обоих из них.

Это позволяет отметить, что для рекомендательных систем можно использовать оба подхода, но стоит учесть, что подход с сингулярным разложением матриц обладает меньшей точностью.