Лабораторная работа №14. Рекомендательные системы.

В этой лабораторной работе вам необходимо создать систему рекомендаций для фильмов с использованием алгоритма коллаборативной фильтрации. Теоретический материал для лабораторной работы можно найти в разделе 11 учебного пособия.

1 Загрузка и отображение данных

На первом шаге работы вам не нужно писать собственного кода. Из файла ex8movies.npy загружаются исходные данные:

num_users – количество пользователей системы (n_u в теоретическом материале);

num_movies – количество фильмов в системе (n_m в теоретическом материале);

num_features – количество признаков фильма (n);

Y – матрица оценок фильмов пользователями размером $n_m \times n_u$;

R – матрица признаков наличия оценки фильма пользователем размером $n_m \times n_u$;

X – матрица признаков фильмов размером $n_m \times n$;

Theta – матрица параметров пользователей размером $n_u \times n$ (Θ);

movies – список названий фильмов длиной n_m.

Данные для этой лабораторной работы взяты из базы MovieLens 100K (https://grouplens.org/datasets/movielens/).

Исходными данными для обучения коллаборативной фильтрации являются матрицы *Y* и *R*, тогда как *X* и Θ получаются в результате обучения модели. Тем не мнее из исходных данных загружаются предобученные *X* и Θ для того, чтобы сравнить с полученными вами результатами и оценить правильность работы программы.

Удалите оператор return, чтобы перейти к следующему шагу.

2 Нормализация рейтинга

В алгоритме коллаборативной фильтрации рекомендуется выполнить нормализацию матрицы оценок по среднему значению.

Найдите в программе функцию normalize_ratings и дополните ее необходимым кодом.

Функция принимает на вход матрицы *Y* и *R* и должна вернуть матрицу нормализованных оценок *Ynorm* (того же размера, что и *Y*) и вектор средних значений оценок по каждому фильму *Ymean* (длиной n_m).

Учтите, что при вычислении средней оценки фильма нужно учитывать оценки только тех пользователей, для которых $r(i, j) = 1$.

Когда функция готова, запустите программу. Она вызывает функцию для *Y* и *R* и выводит среднюю оценку первого фильма из выборки. Если функция реализована правильно, полученный результат должен совпадать с ожидаемыми.

Удалите оператор return, чтобы перейти к следующему шагу.

3 Функция стоимости без регуляризации

На третьем шаге необходимо реализовать вычисление функции стоимости коллаборативной фильтрации, чтобы потом можно было обучить рекомендательную модель. На данном шаге вы реализуете функцию стоимости без регуляризации, далее на шаге 5 будет добавлена регуляризация.

Найдите в программе функцию cofi_cost и дополните ее необходимым кодом для вычисления стоимости. Функция принимает параметры:

params – вектор параметров рекомендательной модели (подробнее далее);

Y – матрица оценок фильмов пользователями;

R – матрица признаков наличия оценки фильма пользователем;

num_users – количество пользователей системы;

num_movies – количество фильмов в системе;

num_features – количество признаков фильма;

lamb – параметр регуляризации (на данном шаге передается равным нулю).

Функция должна вернуть значение *J* стоимости от переданных параметров.

Вектор параметров *params* составлен из декомпозированных матриц *X* и *Theta* таким образом, что сначала в вектор записывается первая строка матрицы *X*, затем вторая строка матрицы *X* и т.д. до последней строки, далее таким же образом записываются все строки матрицы *Theta*. Длина вектора *params* составляет $(n_m n+n_u n)$.

Подобные подход применялся в лабораторной работе №8 для формирования вектора параметров из матриц весов нейронной сети. Рекомендуется реализация этой функции в матричном виде, чтобы обеспечить высокую производительность.

Когда функция готова, запустите программу. Она вызывает функцию для тестовых значений. Если функция реализована правильно, полученный результат должен совпадать с ожидаемым.

Удалите оператор return, чтобы перейти к следующему шагу.

4 Функция вычисления градиента без регуляризации

На данном шаге вам нужно реализовать вычисление градиента коллаборативной фильтрации. Функция вычисления градиента в совокупности с функцией стоимости позволяют обучить модель коллаборативной фильтрации с использованим градиентного метода оптимизации. На этом шаге вам нужно вычислить градиент без регуляризации.

Найдите в программе функцию cofi_gradient и дополните ее необходимым кодом для вычисления градиента. Функция принимает параметры, аналогичные функции стоимости, описанные на предыдущем шаге работы: *params*, *Y*, *R*, *num_users*, *num_movies*, *num_features*, *lamb*.

Функция должна вернуть вектор градиента *grad* той же длины, что и *params*. Обратите внимание, что частные производные по $x^{(i)}$ и по $\theta^{(j)}$ вычисляются по разным формулам. Полученные частные производные должны быть также декомпозированы в один вектор *grad*. Причина этих преобразований заключается в том, что библиотечные функции оптимизации предназначены для работы с вектором параметров и не могут оптимизировать матрицы произвольного размера.

Также рекомендуется реализация этой функции в матричном виде, чтобы обеспечить высокую производительность.

Когда функция готова, запустите программу. Она вызывает функцию для тестовых значений, затем вычисляет градиент численным методом (см. раздел 5.8 учебного пособия) и определяет величину расхождения между векторами градиента, вычисленными двумя разными способами. Если функция реализована правильно, расхождение должно быть мало (менее 10^{-9}).

Удалите оператор return, чтобы перейти к следующему шагу.

5 Функция стоимости с регуляризацией

На этом шаге необходимо добавить регуляризацию к вычислению функции стоимости коллаборативной фильтрации. Вернитесь к функции cofi_cost и дополните ее необходимым кодом.

Когда функция готова, запустите программу. Она вызывает функцию для тестовых значений. Если функция реализована правильно, полученный результат должен совпадать с ожидаемым.

Удалите оператор return, чтобы перейти к следующему шагу.

6 Функция вычисления градиента с регуляризацией

На этом шаге необходимо добавить регуляризацию к вычислению вектора градиента коллаборативной фильтрации. Вернитесь к функции cofi_gradient и дополните ее необходимым кодом.

Когда функция готова, запустите программу. Она также вызывает функцию для тестовых значений, вычисляет градиент численным методом и определяет величину расхождения между вычисленными двумя способами векторами градиента. Если функция реализована правильно, расхождение должно быть не больше ожидаемого.

Удалите оператор return, чтобы перейти к следующему шагу.

7 Обучение коллаборативной фильтрации

Теперь можно приступить к обучению рекомендательной модели коллаборативной фильтрации.

Согласно алгоритму, при обучении модели на основе матрицы оценок пользователей вы должны получить матрицу признаков фильмов X и матрицу параметров пользователей Θ.

Найдите в программе функцию cofi_train и дополните ее необходимым кодом для обучения модели. Функция принимает параметры:

Y – матрица оценок фильмов пользователями;

R – матрица признаков наличия оценки фильма пользователем;

num_users – количество пользователей системы;

num_movies – количество фильмов в системе;

num_features – количество признаков фильма;

lamb – параметр регуляризации.

Функция должна вернуть:

X – матрица обученных признаков фильмов размером $n_m \times n$;

Theta – матрица обученных параметров пользователей размером $n_u \times n$.

Для минимизации функции стоимости и нахождения параметров X и Θ, при которых достигается минимум, можно воспользоваться функцией **minimize** из библиотеки **scipy.optimize**. Пример ее использования можно найти в предыдущих лабораторных работах №5 и №9. Документация по функции имеется на официальном сайте https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html.

Рекомендуется использовать метод TNC (усеченного градиента Ньютона) или CG (сопряженных градиентов) для данной задачи.

Обратите внимание, что в алгоритме коллаборативной фильтрации требуется начальная инициализация признаков и параметров небольшими случайными значениями перед началом обучения. Также учтите, что матрица оценок нормализована с использованием написанной вами ранее функции normalize_ratings.

Когда функция обучения готова, запустите программу. Рекомендательная система обучается с использованием загруженной из файла и затем нормализованной матрицы оценок, в результате получаются обученные матрицы X и *Theta*. С использованием этих матриц вычисляется предсказание оценок пользователей для всех фильмов.

Далее для проверки выдается рекомендация для первого пользователя с 10-ю фильмами, которые он ранее не смотрел (не оценивал) и для которых предсказаны наиболее высокие оценки.

Вы должны получить следующий список фильмов (порядок может отличаться):

Lawrence of Arabia (1962)
Great Day in Harlem, A (1994)
They Made Me a Criminal (1939)
Marlene Dietrich: Shadow and Light (1996)
Star Kid (1997)
Entertaining Angels: The Dorothy Day Story (1996)
Saint of Fort Washington, The (1993)
Aiqing wansui (1994)
Someone Else's America (1995)
Prefontaine (1997)

На этом выполнение лабораторной работы №14 завершается.