1 Рекомендательные системы

Сегодня рекомендательные системы встречаются повсеместно. В интернет-магазине вы можете увидеть блоки с «похожими товарами», на новостном сайте «похожие новости» или «новости, которые могут вас заинтересовать», на сайте с арендой фильмов это могут быть блоки с «похожими фильмами» или «рекомендуем вам посмотреть».

Задача рекомендательной системы залючается в нахождении небольшого числа фильмов (Item), которые скорее всего заинтересуют конкретного пользователя (User), используя информацию о предыдущей его активности и характеристиках фильмов.

Широко известен конкурс компании Netflix, которая в 2006 году предложила предсказать оценки пользователя для фильмов в шкале от 1 до 5 по известной части оценок. Победителем признавалась команда, которая улучшит RMSE на тестовой выборке на 10% по сравнению с их внутренним решением. За время проведения конкурса появилось много новых методов решения поставленной задачи.

Обычно в таких задачах выборка представляет собой тройки $(u, i, r_{u,i})$, где u – пользователь, i – фильм, $r_{u,i}$ – рейтинг. Далее будем считать, что рейтинги нормализованы на отрезок $[0, 1]$.

2 Neighborhood подход в коллаборативной фильтрации

Имея матрицу user-item из оценок пользователей можно определить меру adjusted cosine similarity похожести товаров i и j как векторов в пространстве пользователей:

$$sim(i,j) = \frac{\sum_{u \in U}(r_{u,i} - \overline{r_u})(r_{u,j} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,i} - \overline{r_u})^2}\sqrt{\sum_{u \in U}(r_{u,j} - \overline{r_u})^2}} \tag{1}$$

где U – множество пользователей, которые оценили фильмы i и j, $\overline{r_u}$ – средний рейтинг пользователя u.

Рейтинги для неизвестных фильмов считаются по следующей формуле:

$$\hat{r}_{u,i} = \frac{\sum\limits_{j:r_{u,j} \neq 0} sim(i,j) r_{u,j}}{\sum\limits_{j:r_{u,j} \neq 0} sim(i,j)} \tag{2}$$

Такой подход называется item-oriented. Обратим внимание на то, что $sim(i,j) \in [-1, 1]$. Это может привести к делению на ноль или значениям $\hat{r}_{u,i}$ вне диапазона $[0, 1]$. Избавиться от этой проблемы можно, например, положив равными нулю отрицательные значения $sim(i,j)$.

3 Описание задания

В рамках данного задания Вам будет необходимо реализовать коллаборативную фильтрацию по формулам 1, 2 с использованием фреймворка MapReduce. Ваша программа, получая на вход список троек $(u, i, r_{u,i})$ и список соответствий между номером фильма и его названием, должна вывести для каждого пользователя топ-3 его самых рейтинговых фильма.

В качестве датасета предлагается использовать MovieLens. Для отладки своей программы Вы можете использовать «small» версию датасета, однако, для получения оценки за данное задание Вы должны будете предоставить результат работы своей программы на «full» версии.

Для выполнения задания Вам могут пригодиться следующие ссылки:

- Запуск Hadoop кластера на платформе Google Cloud
- Документация по Hadoop Streaming
- Пример запуска Hadoop Streaming программы на кластере