Assignment 2: Report

Practical Machine Learning and Deep Learning Assignment 2 - Movie Recommendation

---

Adela Krylova
a.krylova@innopolis.university

Introduction

The main goal of the recommendation system is to provide a couple of movies which a user might like based on the ratings this person gave to other movies. This is a typical task of **collaborative filtering**. To solve this task I decided to use a well known algorithm **Alternating Least Squares (ALS)** from Pyspark. As an evaluation metric I used **RMSE**. Overall, on testing my model reached **0.836 RMSE** (tested on 30% data from the provided dataset).

Analysis

Building my solution I used **2 datasets**. Both datasets are provided by **GroupLense**. All the datasets by GroupLense can be found here. In ALS it is important to find the right parameters, for such a purpose I use the smaller movielens datasets which contain only 100K ratings. For the final training I used the ml-latest dataset, containing 33,000,000 ratings. To make it possible to process this amount of data I used **Pyspark**.

During the data processing part it was necessary to load all the data appropriately into **Spark RDD** (resilient distributed dataset). Also for the training I use only the rating of movies and I omit any demographic information about the user, since we do not need such information in ALS.

I used **30/70 split** for test/train and the full small dataset for parameter (rank) tuning. The test data are in format:

user_ID, movie_ID, rating. You can create your own file for a user and make a prediction using this structure.

Model Implementation

Alternating least squares (ALS) is a popular algorithm for **collaborative filtering**. In ALS the users and movies are represented as a **table R**. This table contains ratings of movies by users (other information is omitted). However, since not all users saw all movies some cells are blank. The task could be reformulated as "**finding an optimal way to fill in these blanks**". ALS works based on the **assumption** that if person 1 likes movie B and C, then person 2 who likes movie B and has not seen movie C, will like C. **ALS basically splits the 2D matrix R into its multiplying vectors Q and P and tries to find the best fitting values to fill the blanks in original R.** An important parameter in ALS is a rank - rank means how much information must be considered. For example in my solution the optimal **rank is 4** - that means I use, user_ID, movie_ID, rating and a time_stamp. Other features seem to be redundant.

Model Advantages and Disadvantages

ALS is an **on-line algorithm** that means that the algorithm can process data serialy, that is a huge advantage while working with big data and movie recommendation system is one of typical challenges for **Big Data**.

ALS using Pyspark is **easy to implement** and works fine. Moreover, considering the vast amount of data, it worked quite fast. The complexity is **linear** per example.

One of the disadvantages is **hyperparameter sensitivity**; however, in this task it was not such a big problem.

Another problem might be if there is a **lack of data**, but in this task we had enough data.

ALS uses **big matrices and multiplication of them**, in some tasks it might result in usage of too much memory. For that reason I used Pyspark for training.

Training Process

As already described the training consists of **two stages - 1st parameter tuning (rank) and the training itself**. I used the smaller dataset for finding the optimal rank and the full dataset for training. **Actually the solution predicts the ratings of movies for users who have not seen the movies yet. For recommendation purposes movies with TOP-N ratings could be selected.**

Evaluation

As a metric I use **RMSE** (root mean square error). The RMSE calculates the difference between predicted rating of the movie and the actual rating.

```
For testing data the RMSE is 0.8261327073910048
```

Results

Overall, I managed to meet the requirements. The **RMSE 0.83** could be considered as a **good result**.

References

My solution was built based on a lab assignment from a Big Data course which I took during my exchange semester at ITS, Indonesia.