**Distributed Item Based Recommendation Algorithm**

Project Objective

To implement a distributed Hadoop based Item recommendation algorithm

Project Overview

Recommendation algorithms can be used by online retailers or any large company to recommend items/products to customers based on their past buying history. By using recommendation customers can buy items which they like increasing the revenues for the company. The purpose of the project is to build a recommendation algorithm using Hadoop and MapReduce. Since the data is huge we can leverage the massively parallel computation of Hadoop and MapReduce for processing the data.

The recommendation algorithm is based on Collaborative filtering technique which states that if a person “ABC” has the same opinion as a person “CBA” on one particular issue, person ABC will most likely have the same opinion on another issue as of CBA compared to any other person chosen randomly in a given set.

The algorithm is based on the details given in Chapter 6 of “Mahout in Action Book”.

Apache Mahout’s machine learning introduced a co-occurrence matrix. For N items, it creates an N\*N square matrix. Each row/column represents an item, and an element in the matrix represents co-occurrence value between the row and column items. It then introduced a user vector to show the preference of users to these objects according to the items they bought before. By multiplying the co-occurrence matrix with user’s preference vector, the algorithm produces a vector that leads to recommendations

Algorithm Description

The first step is construct a N \* N item based matrix as shown below as an example,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Item1 | Item2 | Item3 | Item4 | Item5 |
| Item1 | 5 | 2 | 3 | 0 | 5 |
| Item2 | 2 | 3 | 2 | 1 | 4 |
| Item3 | 3 | 2 | 3 | 2 | 1 |
| Item4 | 0 | 1 | 2 | 3 | 4 |
| Item5 | 5 | 4 | 1 | 4 | 3 |

The values in the cells corresponds to the nbr of times item occurring in the users ratings together. From the above given matrix, we can conclude that item1 and item3 were rated by 3 users.

By multiplying the above matrix with the user matrix we can recommend the top items to the users as shown below,

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Item1 | Item2 | Item3 | Item4 | Item5 |  | User1 |  | Matrix Multiplication |
| Item1 | 5 | 2 | 3 | 0 | 5 |  | 0 |  | 19 |
| Item2 | 2 | 3 | 2 | 1 | 4 |  | 2 |  | 16 |
| Item3 | 3 | 2 | 3 | 2 | 1 |  | 5 |  | 19 |
| Item4 | 0 | 1 | 2 | 3 | 4 |  | 0 |  | 12 |
| Item5 | 5 | 4 | 1 | 4 | 3 |  | 0 |  | 13 |

The user matrix can be constructed based on the ratings user gave to items. If the user hasn’t rated an item, the rating is considered as 0.

The matrix multiplication can be obtained by multiplying the row1 with the user1 row. After the matrix multiplication the highest value items which have not be rated can be recommended to users. In the above example item1 can be the top recommended item for user1.

Input Source

The freely available movie lens database can be used for calculating the matrix and the user recommendation items.

<http://files.grouplens.org/datasets/movielens/ml-100k/>

HPC ABDS Technologies Used in the project

Cloudera Hadoop – CDH 5.2 5 Node cluster

MapReduce

Yarn (Resource management)

Java

Maven (Build Tool)

Spring Tool Suite (For development)

Hue (For HDFS file browse and job submission)

OOZIE (For jobs execution)

Development

Below table shows the creation of 5 map reduce jobs used in the algorithm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Job Description | Input | Output | Details |
| 1 | Generating User vectors from the input dataset | Movie lens dataset where the dataset is in the format of userid, itemid, rating | UserId ItemId:Rating; ItemIdRating | This will generate the user vectors with all the items rated by a user |
|  |  |  |  |  |
| 2 | Generating Item Vectors from the user vectors | User Vectors from Job1 | ItemId Userid:Rating;UserId:Rating | This will generate the item vector with all the users rated a given item |
|  |  |  |  |  |
| 3 | CoOccurrence Vector | User Vectors from Job1 | ItemId ItemId:Coocuurence;….. | This will generate the Cooccurrence vector with the cooccurrence value which shows the two items appearing together in the users ratings. |
|  |  |  |  |  |
| 4 | MergingVector | CoOccurrence Vector from Job3 and Item Vector from Job2 | ItemId ItemId:CoOccurrence and UserId:Rating | This will combine both the item vector and the cooccurrence vector and produces a single record |
|  |  |  |  |  |
| 5 | MatrixProduct | Vector from Job4 | UserId ItemId, ItemId | This job will perform the matrix multiplication required by the algorithm. The output of this will produce the user with the list of recommenced items. |

Reference

1. I-590 Big Data Software and Projects ClassRoom Videos
2. Mahout in Action Book
3. And many references online about MapReduce