SMAI Interim Eval 2

Team 10

Problem Introduction

Given the user’s preferences in movies and a general dataset of all the users and their ratings we have to predict how the user would rate an arbitrarily selected movie. To solve this problem we are using collaborative filtering which is a mechanism to filter massive amounts of data based upon a previous interactions of a large number of users. In this project we will analyze and benchmark several collaborative filtering methods. These methods are :- Stochastic Gradient Descent (SGD), Alternating Least Squares (ALS), Biased Stochastic Gradient Descent (B-SGD), Weighted Alternating Least Squares (W-ALS).

Work Completed

We have implemented Alternating Least Squares (ALS) and Weighted Alternating Least Squares (W-ALS) for Interim Evaluation 2.

By the final evaluation we would have completed the other two algorithms Stochastic Gradient Descent (SGD) and Biased Stochastic Gradient Descent (B-SGD).
We will be comparing the various algorithms as well.

Problem Challenges

- Sparsity of user preference and Scalability in computation
  User preference sparsity makes it hard to find users who share similar taste
  Large sparse matrices make hard to do computation in general.

- Shilling Attack
  People who like one brand may give consistently high scores to that brand and give poor scores to the competitors no matter what their real experience are.

Dataset Information

The dataset downloaded from :- http://grouplens.org/datasets/movielens/1m
contains around 1,000,209 ratings of approximately 3,900 movies
made by 6,040 MovieLens users who joined MovieLens in 2000.

The dataset is separated into three tables - ratings, users, movies.

- Ratings contain UserID, MovieID, Rating (1-5).
- User table contains UserID, Gender, Age Group and Occupation.
- Movies table contains MovieID, Title and Genres.

We will add genre and age group information from Movies and User respectively, if time permits later.

Features Explanation

RATINGS

UserID::MovieID::Rating::Time stamp

- UserIDs range between 1 and 6040
- MovieIDs range between 1 and 3952
- Ratings are made on a 5-star scale (whole-star ratings only)
- Timestamp is represented in seconds
- Each user has at least 20 ratings

USER

UserID::Gender::Age::Occupation::Zip-code

- Gender is denoted by a "M" for male and "F" for female
- Age is chosen from the following ranges:
    * 1: "Under 18"
    * 18: "18-24"
    * 25: "25-34"
    * 35: "35-44"
    * 45: "45-49"
    * 50: "50-55"
    * 56: "56+"

MOVIES

MovieID::Title::Genres

* Action
* Adventure
* Animation
* Children's
* Comedy
* Crime
* Documentary
* Drama
* Fantasy
* Film-Noir
* Horror
* Musical
* Mystery
* Romance
* Sci-Fi
* Thriller
* War
* Western

Performance Metrics

- As we are creating a rating system the output is purely subjective there is no right or wrong.
- However we have defined an error between what the user gave and what we predicted that the user will give
- The error function we are using is RMSE given by :-

$$RMSE = \sqrt{\frac{1}{n}\sum_{u,v}|(p_{u,v} - r_{u,v})^2|}$$

Here, $p_{u,v}$ is the predicted rating and $r_{u,v}$ is the observed rating of user u and movie v respectively. $p_{i,j}$ is the inner product of ui and vj.

$$(u_i, v_j) = \min_{u,v} \sum_{(i,j)\in K} (p_{i,j} - r_{i,j})^2 + \lambda(||u_i||^2 + ||v_j||^2)$$

The low rank approximation problem learns the factor vectors (u_i, vj). In order to avoid overfitting a technique known as Tikhonov regularization is used to transform the low rank approximation problem into the second equation given.

Algorithm 1: Alternating Least Squares

Steps:

1.) Initialize matrix V by assigning the average rating for that movie as the first row, and small random numbers for the remaining entries.

2.) Fix V , solve U by minimizing the RMSE function.

3.) Fix U, solve V by minimizing the RMSE function similarly.

4.) Repeat Steps 2 and 3 until convergence.

Algorithm 2 : Weighted Alternating Least Squares

Weighted alternating least squares is an algorithm very similar to alternating least squares but with a different objective function. W-ALS is aimed at trying to optimize collaborative filtering algorithms for datasets that are derived off of implicit ratings. W-ALS introduces different confidence levels for which items are preferred by the user changing the objective function to be minimized to.

$$(u_i, v_j) = \min_{u,v} \sum_{(i,j)\in K} c_{i,j}(p_{i,j} - r_{i,j})^2 + \lambda(||u_i||^2 + ||v_j||^2)$$

$p_{i,j}$ and $r_{i,j}$ are the predicted rating and observed rating of user i and movie j respectively.

$$c_{i,j} = 1 + \alpha log(1 + r_{i,j}/\epsilon)$$

This is the algorithm we used to compute the confidence level. Here ci,j measures the confidence of the predicted preference $p_{i,j}$.

Analysis and Result

We split the ratings data into 90% training data and 10% testing data. Here is the preliminary analysis :-

- ALS mean error comes to be around 2.6 after 100 iterations and it took 10minutes. The testing error was around 3.2.
- W-ALS mean comes around 1.07 after 10 iterations and it took 2 days. Here the testing error was around 1.45