Final Project Report

Gurpreet Singh

Introduction

In this project, we build a movie recommendation system. The system is able to make two kinds of recommendations non-personal recommendation and personal recommendation. For e.g. For a user, which is new to the system, the systems will use recommend him the most popular movies in the system and then analyze his behavior. For all the existing users, the system will make the personalized recommendations. In this project I also did the comparison study of few algorithms, which are mostly used in recommendation systems for e.g: nearest neighbor method, kmeans clustering, SVD (ALS Alternate least squares) and deep neural networks. I used movie lens 100K dataset for training and the testing purpose., which is available from the university of Illions Urbana Champaign.

Problem definition

Recommendation systems are the system which recommend the item to a user based upon his previous behavior.

The problem of the recommendation can be defined as:

Given n(u) = number of users

n(m) = number of items.

R (i, j) =1 if user i has rated item j

Y (i, j) = rating given by user i to item j

Theta(i)= feature vector for user i.

X(j)= feature vector for item j

For user i, given item j, predict rating r (i, j)

The accuracy of the model is determined by using mean square error (MSE)

$$\mathbf{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y_i} - Y_i)^2$$

Where n is the total number of ratings predicted. Y_hat is the predicted rating and Yi is the actual rating. So, Let M1, M2, M3, M4 are the mean square errors of different models. The model with lowest MSE is the best model.

Motivation

Recommendation systems are nice blend of analytics, statistics and machine learning. Recommendation systems are widely used in almost all the fields in industry being from financial to entertainment. There are two kinds of recommendation systems, content based and model based.

However, the scalability was always an issue with content based systems and finding an adaptable and good model is a research problem. Deep learning has been very successful in solving NLP and computer vision problems. However, researchers are trying to figure out, how we can use state of the art deep learning for recommendation systems. So, I decided to implement a movie recommendation systems with state of the art ALS (Alternate least square), KNN, K means method and I also tried to deep learning on this system. In the end, I did the comparison study of the all these algorithms.

Related work

Researchers has been working on recommendation system even before the evolution of the internet. But with the evolution of data based technologies this field gained lots of success. Group lens used the Pearson coefficient formula for user based collaborative filtering. Which worked well with content based systems. But with the problem of scalability and handling new user, researchers moved to the model based algorithms. SVD and ALS are most popular model based algorithms widely used in industry. Recently, few researchers have made the success to apply deep learning technologies using auto encoders to reduce the root mean square error below the ALS method. I have referenced their work and tried to reconstruct the model using torch and Movielens dataset.

Methods

I tried following methods for predicting a rating of the movie.

a) User-User Collaborative Filtering(k-NN)
b) K means clustering
c) ALS (Alternate least square)
d) Collaborative filtering using Neural networks using auto encoders

Nearest Neighbor:

The user-user collaborative filtering is also known as K-NN collaborative filtering algorithm. This method is pretty straight forward, which tries to find out the users which behaves similar to the current user and use their ratings to predict the rating of the current user for the given movie. There are various similarity methods which can be used to find out the similarity between users for eg: linear distance, cosine distance and the Pearson co-efficient distance. In our implementation we tried Pearson co-efficient to measure the similarity between two users.

$$s(u,v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}}$$

Here S(u,v) represents the similarity between user u and user v.To calculate s, normalize the ratings of the user u and user v by subtracting the mean value of the ratings given by respective user from all the ratings and then take product of the sum of the ratings by the respective for the items which are rated by both the user u and user v. Followed by, we divide it by the product of the square root of sum of squared ratings.

The rating for the particular user is predicted by using the weighted mean of the ratings of the K top users.

K means clustering and Recommendation

K means clustering algorithm cluster the similar users and then I tried to recommend the movies of one user to other users and predicted how other user rates that particular movie. Every time a new user comes in its label is predicted using its distance from the centroid of the clusters and the user is recommended with the items from the movies which are watched and highly rated by members of that cluster. The loss is calculated by using root mean square error of the actual rating and the predicted movie rating.

```
1. Initialize centroids $\mu_1, \ldots, \mu_k \in \mathbb{R}^n$ randomly.
2. Repeat until convergence : {
        For every $i$, set
            $c^{(i)} := \arg\min_j \|x^{(i)} - \mu_j\|^2$
        For each $j$, set
            $\mu_j := \frac{\Sigma_{i=1}^m 1\{c^{(i)}=j\}x^{(i)}}{\Sigma_{i=1}^m 1\{c^{(i)}=j\}}$
}
```

Users in the complete dataset were labeled based on cluster assignment.

ALS (Alternate Least Square)

ALS is a model based algorithm. It tries to learn the parameters which best fit the function f(x) such that for all x in x1, x2 , x3…… the loss is minimized. The ALS method is tries to minimize the sum of the squared error (SSE) ie for given dataset the ALS method will try to find out parameters such that sum of all the squared error is minimized.

$$S = \sum_{i=1}^{i=n} r_i^2,$$

$$r_i = y_i - f(x_i)$$

The cost function for the ALS is given below. Here J(xu) represents the cost value for the user feature vector and J(yi) represents the cost function for the item feature vector.

$$J(x_u) = (q_u - x_u Y)W_u(q_u - x_u Y)^T + \lambda x_u x_u^T$$

$$J(y_i) = (q_i - Xy_i)W_i(q_i - Xy_i)^T + \lambda y_i y_i^T$$

Taking derivative of it, our objective function is given by:

$$x_u = (YW_uY^T + \lambda I)^{-1}YW_uq_u$$

$$y_i = (X^TWiX + \lambda I)^{-1}X^TW_iq_i$$

Here lambda is the regularizer parameter to prevent from overfitting.

Neural Networks Using Auto Encoders:

Neural networks are based upon the brain model in which multiple neural units work together to solve a problem. The autoencoders is a neural network whose goal is to use unsupervised learning to learn representation of the data. It is widely used in dimensionality reduction and generating data samples. In our model, we try to generate the missing ratings using autoencoders.

$$L_{2,\alpha,\beta}(\mathbf{x}, \tilde{\mathbf{x}}) = \alpha \left(\sum_{j \in \mathcal{J}(\tilde{\mathbf{x}}) \cap \mathcal{K}(\mathbf{x})} [nn(\tilde{\mathbf{x}})_j - x_j]^2 \right) + \beta \left(\sum_{j \notin \mathcal{J}(\tilde{\mathbf{x}}) \cap \mathcal{K}(\mathbf{x})} [nn(\tilde{\mathbf{x}})_j - x_j]^2 \right)$$

Here, K(x) is index of known values for x. and L represents the loss function for the given equation. The alpha and beta represents the rates at which the model tries to clean the corrupted data and generate ratings respectively.

Evaluations:

I used MSE (Mean square error) as a measure to compare different algorithms. Based upon these values the Result matrix is given below.

Method	Mean square error		
KNN	1.2		
KMeans	1.5		
ALS	0.92		
NN (Autoencoders)	.7764		

References:

Hybrid Recommender System based on Autoencoders Florian Strub Univ. Lille, CNRS, Centrale Lille, Inria UMR 9189 - CRIStAL F-59000 Lille, France florian.strub@inria.fr

A Music Recommendation System with a Dynamic K-means Clustering Algorithm Dong-Moon Kim1 , Kun-su Kim1 , Kyo-Hyun Park1 , Jee-Hyong Lee1 and Keon Myung Lee2 1 Department of Electrical and Electronic Engineering, SungkyunKwan University, Korea 2 School of Electrical and Computer Engineering, Chungbuk National University, Korea 1 {skyscrape, kkundi, megagame}@skku.edu , 1 jhlee@ece.skku.ac.kr, 2 kmlee@cbnu.ac.kr

Collaborative Filtering Recommender Systems By Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan