MEF UNIVERSITY

MATRIX FACTORIZATION FOR RECOMMENDATION ENGINE

Capstone Project

Erol Yaldır

İSTANBUL, 2019

MEF UNIVERSITY

MATRIX FACTORIZATION FOR RECOMMENDATION ENGINE

Capstone Project

Erol Yaldır

Advisor: Ph.D Alper Öner

İSTANBUL, 2019

MEF UNIVERSITY

Name of the project: Matrix Factorization For Recommendation Engine
Name/Last Name of the Student: Erol Yaldır

I hereby state that the graduation project prepared by Erol Yaldır has been completed under my supervision. I accept this work as a "Graduation Project".

20/12/2019
Ph.D Alper Öner

I hereby state that I have examined this graduation project by Erol Yaldır which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

20/12/2019

Director
of
Information Technologies Program

We hereby state that we have held the graduation examination of and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Ph.D Alper Öner.....	……..
2. ……..	……..

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Name Date Signature

EXECUTIVE SUMMARY

MATRIX FACTORIZATION FOR RECOMMENDATION ENGINE

Erol Yaldır

Advisor: Ph.D Alper Öner

AUGUST, 2019, Number pages (e.g 16 pages)

The size of stored data increase day by day.For a research 2.5 quintillion bytes of data produce in a single day(Irfan A. 2018).Because of producing huge info,Information technologies involve rapidly.Recommendation system or engine is a branch of information technologies that filtering data that seeks to prediction for product,rate or disease diagnosis for users,customers or patients.Mainly recommendation systems are used for commercial applications.

Recommendation systems are seen very different areas, for offering a different product in e-commerce such as Amazon or eBay,friend or content recommender for social media as Instagram,Twitter or Facebook or most commonly prediction rating and offer playlist of video or music services like YouTube,Spotify or Netflix.

While machine or deep learning is main way of recommendation systems , it is not the single only solution.There are lots of ways to build up recommendation system.Simply if have few data or want to build a minimal and fast. Simple offer a product or solution to customer,based on that user did before,we can simply build an system that the other users did before.So we can prefer classification,collaborative filtering,popularity based,Nearest neighbor or Matrix factorization.

Matrix factorization most basic model that based on collaborative filtering and is also one of the most important technique in recommendation system. For example in movie recommendation system that also i will use in this article also,if a user rate a movie that saw before from one to five,this collection of feedback can be represented in a form of a matrix.Each row represents each users and also each column means different movies.Because of all users didn't watch all movies,our matrix will be sparse.

In this article we will compare the matrix factorization with traditional recommendation systems.

Key Words: Recommendation, System, Information Technologies, Collabrative Filtering , Content-Based, Matrix factorization

ÖZET

ÖNERİ SİSTEMLERİNDE MATRİS ÇOĞALTMA YÖNETMİ

Erol Yaldır

Tez Danışmanı: Dr. Alper Öner

AĞUSTOS, 2019, sayfa sayısı (ör. 16 sayfa)

Saklanan data boyutu her geçen gün artmaktadır.Bir araştırmaya göre bir günde 2.5 kentilyon byte data üretilip saklanmaktadır.Bu kadar büyük datanın üretilmesinden dolayı,bilgi teknolojileride hızla gelişmektedir.Bilgi teknolojilerinin bir dalı olan öneri sistemleri yada motorları,filtrelenmiş datalarla kullanıcılara,müşterilere yada hastalara,ürün , oran yada hastalık sonuçları üzerine öneride bulunmaktadır.Genel olarak öneri sistemleri ticari olarak kullanılmaktadır.

Öneri sistemleri çok farklı alanlarda görülebilir.Amazon yada eBay da farklı ürünler önerirken,Instagram, Twitter yada Facebook da içerik yada arkadaş önerirken yada en genel olarak içerik ve playlist önerisi için video ve müzik servis sağlayıcıları olan YouTube,Spotify ve Netflix de kullanılmaktadır.

Derin yada makine öğrenmesi öneri sistemleri için birincil yol olarak gözükmesine rağmen tek değillerdir.Öneri sistemi gelişmesi için bir çok yol bulunmaktadır.Basit olarak eğer az datamız olabilir ve hızlı bir şekilde oluşturmak isteyebiliriz.Basit bir şekilde ürün yada çözüm önerisi için,müşterinin daha önce yaptığı şeyi,diğer müşterilerin yaptığı ile kıyaslarız.Böyle bir sistem için sınıflandırma,ilişkisel filtreleme,popülariteye dayandan,En yakın Komşu yada Matris çoğaltma yöntemi kullanabiliriz.

İlişkisel filtreleme yönetemine dayanan Matris çoğaltma yöntemi,Öneri sistemlerinin en temel ve en önemlilerinden birisidir.Örnek olarak bu makalede de kullanacağımız film öneri sisteminde,eğer bir kullanıcı izlediği bir filmleri birden beşe kadar oylarsa,bu geri bildirim koleksiyonu bir matris formunda gösterilir.Her bir satır bir kullanıcıyı temsil ederken ,her bir kolonda bir filme karşılık gelmektedir.Bütün kullanıcılar bütün filmleri izlemediği için datamız az olabilir.

Bu makalede matris çoğaltma yöntemini diğer geleneksel yöntemlerle kıyaslıyacağız.

Anahtar Kelimeler: Öneri sistmeleri,Bilgi teknolojileri,İlişkisel filtreleme,İçerik tabanlı,Matris çoğaltma

TABLE OF CONTENTS

.

1. INTRODUCTION

The reproducing data in the world about any branch of life is growning extremely.For a research 2.5 quintillion bytes of data produce in a single day(Irfan A. 2018).While reproducing huge data,reaching right data is getting harder then harder.So that offering or suggest right data to right user is more important than producing a perfect component.This fact also effected the all sectors on web.For example offering right product to right customer will be better than gave that product cheaper.And also offering a suitable book to a reader will be increase your sell.How can we get right book to right reader?We should look at what did buy our other customer buy before.If we can classified our customer perfectly we can offer their products to each other.

The median length of a top 100 US-grossing films between 1994 and 2015 was 110 minutes.(Stephen F. 2016) This means that if i watched a wrong movie for me i wasted about 110 min.Because of this fact movie recommendation is a getting more important.

With the extreme increasing of computational power,store and work with big data and machine learning algorithm,classificated users rating about watched movie is more easily.With collabrative filtering we will offer highly rated movie to users that have same sanse with others who watched those movie.We will use python language during exams and mainly use pandas,numpy and supriselib for extract dataset set and find best models.

1.1. About the Dataset

The dataset contains about 20 million movie rating that stable benchmark dataset.In this dataset 20 million ratings given by 138k users and 465k tag applications applied to 27k movies.Includes tag genome data with 12 million relevance scores across 1,100 tags.The dataset firstly released in April 2015 and updated on October of 2016.

The dataset doesn't contain any demographic data about user.All of the users only represented by id and there is not any other information about them. There are 6 different csv files these tag,rating,movie,link,genome_scores and relevance.csv files.But we only focus on tag and rating datasets.

2. LITERATURE REVIEW

Because of the importance of recommendation systems,many developers and researchers are thinking on this area on todays.And also getting good results and libraries about recommendation engines.However these are generally on traditional machine learning techniques.On the other hand some of these are about upper level part of matrix factorization.

Shulong Chen and Yuxing Peng (2018) aimed integrate explicit and implicit feedback via matrix-factorization technique.Their development showed that the model effectively improve this recommendation system improve results on MovieLens database on content raking and prediction. The system difficulty of PMF+P+N model is fairly high when apply gradient descent algorithm. This is highly accurate and can propose advance for MF-based CF has opted for an increased and static unified schema. The best case of MF model, known as the local low-grade matrix concept-based model, was proposed.

Ling Luo, Haoran Xie, Yanghui Rao, and Fu LeeWang (2019) aimed to make label data an important source of information that reflects a user's interests, because users use labels to express their focal points and emotions of elements.To determine the negative effects of infrequent recommendation data, another link is used to identify the accuracy and preference of labels.Changes in the interests of the user are taken into account in the proposed common SVD model using temporary information.Ling Luo et ak have worked together three matrices, a list of tools in users, labels, and items. Spare rating the problem of overfitting caused by the data can be alleviated in co-factorization.

Weina Zhang,Xingming Zhang,Haoxiang Wang and Dongpei Chen(2019) purposed that deep matrix factorization depends on variational autoencoder for recommendation algorithm.This technique aimed to improvment of accuracy in sparse data for recommendation system.For getting hiden features of content and users ,they used deep variational networks instead of linear method.For the prediction of unknow rate multiplyed users and content hiden features.And also these hiden features rating bias inserted into account.Based on the network,They optimizated method of the network and improvement DVMF algorithm for sparse CF error.

As a result of these researchs , SVD ,SVD++ and NMF will be give us good results.

3. METHODOLOGY

3.1 Data Loading

Firstly we cleaned data for outliers and missing values.Then upload csv files as datasets with pandas.Our datasets columns listed in below.During examination used Visual Studio Code as IDE.

Rating.csv
userId
movieId
tag
timestamp
Tag.csv
userId
movieId
rating
timestamp

3.2 Data Preprocessing

After prepare datasets and cleaned up , started to work on them.Because of our datasets came form two different csv file ,we merged them in a signle dataset with movieId Beucase of having huge dataset about 20 million records,ordered and sperated by timestamp.

Finally datasets was sperated to test and train sets.For each datasets separated to %70 training sets and %30 for the test.

For calculate and log the accury result, Logger and metrics class were developed also.With Aspect Oriented Programinng after finishing train model method,our metrics class directly logged all of accurcy result to logStash.

3.3 Creating the Model Architecture

In training process SVD,SVD++ and NormalPredictor algorithms used and logged seperatly. The configuration of computer that used during process Intel i7 9750H as cpu, 24gb Ram and 256gb ssd.Because of eleminate unhandeled error , i didn't use my developed and non-tested algorithm library. For SVD ,SVD++ and NormalPredictor algorithms used supriseLib. And for the others used main pandas lib. Each method handled separedly. Because of having huge data ,I didn’t use cross-validation techniques.Also data optimizations were applied.Also for laod dataset from csv file and prepare them in AlgorithCore class , I added DataImporter class.This class mainly read all kind of data and convert it to useful dataset for recommendation engine.

3.4 Training the Model

100 different users rating and movie sense selected seperatly and got results. And then calculated their means and recommendation engine used seperatly. After getting results ,compeared with their watched movie with offered movies.And calculated final results for accurcy.In this examine ,I prefered Top-N method for getting best suitable movies.And also I calculated novelty,diversty and coverage cases .Novelty means average of rank of recommended items,diversty is average similarity score between every possible pair of recommendation and coverage is ratio of users for recommendation above threshold.

```
import random
import numpy as np
from DataImporter import DataImporter
from surprise import SVD, SVDpp,NormalPredictor
from AlgorithmCore import AlgorithmCore
def LoadDataset():
    ml = DataImporter()
    print("Loading movie ratings...")
    data = ml.loadDatasetFromFile()
    print("\nComputing movie popularity ranks")
    rankings = ml.getPopularityRanks()
    return (ml, data, rankings)
np.random.seed(0)
random.seed(0)
#Load dataset
(ml, evaluationData, rankings) = LoadDataset()
#Create Algortihm
evaluator = AlgorithmCore(evaluationData, rankings)
# SVD
SVD = SVD()
evaluator.InsertAlgortihm(SVD, "SVD")
# SVD++
SVDPlusPlus = SVDpp()
evaluator.InsertAlgortihm(SVDPlusPlus, "SVD++")
# Add normal recommendation for calculate
Random = NormalPredictor()
evaluator.InsertAlgortihm(Random, "Random")
# calculating Scores
evaluator.ProcessAlgorithm(False)
evaluator.SampleTopNRecs(ml)
```

Figure 1. Training with suprise lib sample

3.5 Testing Model

For the testing our result i used leave-one-out method that spearate some moveies those from user watched before and try to find them in recommendation result.And calculate ratio.And also average reciprocal,cumulative hit rate and rating hit rate testing methods were applied.But for Top-N recommendation cases leave-one –out gave best results.

Mean Absolute Error (MAE) and Root mean squared error (RMSE) are two of the most common metrics used to measure accuracy for continuous variables.(Human 2016)

With 100k Datasets	**RMSE**	**MAE**	**TIME**
SVD	0.934	0.737	0:00:11
SVD++	0.92	0.722	0:09:03
NormalPredictor	0.91	0.718	0:00:10

With 1M Datasets	**RMSE**	**MAE**	**TIME**
SVD	0.873	0.686	0:02:13
SVD++	0.862	0.673	2:54:19
NormalPredictor	0.923	0.698	0:02:41

Figure 2. Testing Results of Model

4. CONCLUSION

The aim of this experiment compare the matrix factorization with treditional machine learning techniques.With small piece dataset 100k SVD++ gave us best RMSE and MAE result.On the other hand there was a enormus spending time that about 50 times larger than the other.If we think this recommendation system online ,it can't be acceptable.

Hovewer if we look up SVD algorithm result,it gave us second best result and one of the best time result.Offering more quickly moveis online is better than better suitable moveis.If we can coorparate with SVD and co-clustring algortihm together we can get best result.The second phase of this experiment will be this case.

And also we should think data sizes.We only worked with 100k and 1 million sized datasets and the minimum spending time is 3 second in worst result.This is not acceptable on online system.Thinking about on this case is one of the most important think.

REFERENCES

Ling L., Haoran X. & Yanghui R. , LeeWang F.(2019) Personalized recommendation by matrix co-factorization with tags and time information (2019) Expert Systems with Applications Volume 119, 1 April 2019, Pages 311-321

Weina Z. , Xingming Z. & Haoxiang W., Dongpei C.(2019) A deep variational matrix factorization method for recommendation on large scale sparse dataset (2019)Neurocomputing Volume 334, 21 March 2019, Pages 206-218

ShulongChen & YuxingPeng (2018). Matrix factorization for recommendation with explicit and implicit feedback *2018* Knowledge-Based Systems Volume 158, 15 October 2018, Pages 109-117

Irfan A.(2018, June 15). 126 Amazing Social Media Statistics and Facts. Retrievedfrom brandwatch: https://www.socialmediatoday.com/news/how-much-data-is-generated-every-minute-infographic-1/525692/

Stephen F. (2016, January 18).16 Are movies getting longer Retrieved from brandwatch: https://stephenfollows.com/are-hollywood-movies-getting-longer/

Human M.W. (2016, March 23). 16 MAE and RMSE — Which Metric is Better?. Retrieved from brandwatch: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d