Recommender System with Mapreduce - 1

赵敏 老师

扫描二维码关注微信/微博
获取最新IT面试情报及权威解答

微信: ninechapter
知乎专栏: http://zhuanlan.zhihu.com/jiuzhang
微博: http://www.weibo.com/ninechapter
官网: www.jiuzhang.com

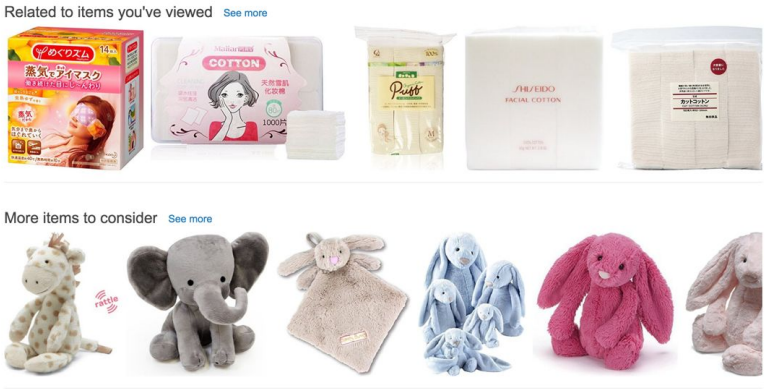
Outline

九章算法

- What is Recommender System
- Why Recommender System
- How to design Recommender System
- Deployment of recommender system on Mapreduce

What is recommender system

- Systems that attempt to predict item that users may be interested in
  - Amazon products recommendation



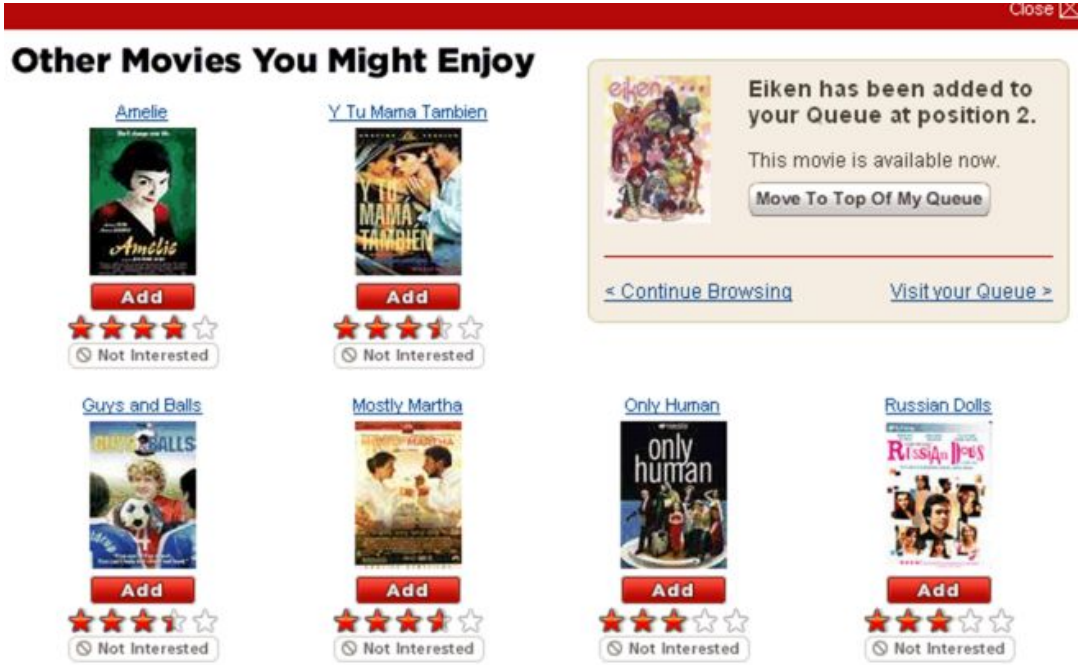
- Systems that help people find information that may interest them
  - Search engine

Why recommender system

九章算法

- Massive MR project
  - Complex Algorithm
  - 5 MapReduce Jobs Chain
  - Put it on your resume!

Design Movie Recommender System



Design Recommender System

九章算法

- Algorithms used for recommender system
- Choose one algorithm then implement it
- Implement recommender system with MapReduce

Algorithms used for recommender system

- User Collaborative Filtering (User CF)
- Item Collaborative Filtering (Item CF)
- ...

Algorithms used for recommender system

九章算法

User CF

- A form of collaborative filtering based on the similarity between users calculated using people's ratings of those items

User	Movie 1	Movie 2	Movie 3	Movie 4
A	10	4	9	9
B	5.5	8	5	5
C	8	2	8.5	recommend

Algorithms used for recommender system

九章算法

Item CF

- A form of collaborative filtering based on the similarity between items calculated using people's ratings of those items

User	Movie 1	Movie 2	Movie 3
A	10	4	9
B	9	5	9
C	8	2	recommend

Algorithms used for recommender system

九章算法

We will use Item CF

- The number of users weighs more than number of products
- Item will not change frequently, lowering calculation
- Using user's historical data, more convincing

Item CF

- Build co-occurrence matrix
- Build rating matrix
- Matrix computation to get recommending result

Item CF

Since this is Item CF, what is the first thing to do?

Item CF

Describe the relationship between different items

Item CF

How to define relationship between different movies

Item CF

Based on user's profile

- watching history
- rating history
- favorite list

Based on movie's info

- movie category
- movie producer

Item CF

We will use rating history to build relationship between movies

- If one user rated two movies, these two are related

Item CF

How to represent relationship between different movies

Item CF

Co-occurrence matrix

A co-occurrence matrix is a matrix that is defined over an image to be the distribution of co-occurring pixel values (grayscale values, or colors) at a given offset.

Item CF

九章算法

User	M1	M2	M3	M4	M5
A	1	1		1	
B	1	1	1		
C		1	1		1
D		1		1	

	M1	M2	M3	M4	M5
M1	2	2	1	1	0
M2	2	4	2	2	1
M3	1	2	2	0	1
M4	1	2	0	2	0
M5	0	1	1	0	1

value(M1,M2) = Count(x_M1 && x_M2) x表示一个人同时看了M1 和 M2 两个电影

Item CF

How to tell the difference between movies towards each user?

User	Transformers	Tiny Times
A	10	4
B	3	9

Item CF

Build rating matrix for each user

Item CF

Rating Matrix

User	M1	M2	M3	M4	M5
A	9	4		8	
B	3	7	8		
C		8	7		4
D		5		8	

Movie	User B rating
M1	3
M2	7
M3	8
M4	0
M5	0

Item CF

Recomend Movie for each user

Item CF

Co-occurrence Matrix

	M1	M2	M3	M4	M5
M1	2	2	1	1	0
M2	2	4	2	2	1
M3	1	2	2	0	1
M4	1	2	0	2	0
M5	0	1	1	0	1

归一化处理 →

	M1	M2	M3	M4	M5
M1	2/6	2/6	1/6	1/6	0
M2	2/11	4/11	2/11	2/11	1/11
M3	1/6	2/6	2/6	0	1/6
M4	1/5	2/5	0	2/5	0
M5	0	1/3	1/3	0	1/3

Item CF

	M1	M2	M3	M4	M5
M1	2/6	2/6	1/6	1/6	0
M2	2/11	4/11	2/11	2/11	1/11
M3	1/6	2/6	2/6	0	1/6
M4	1/5	2/5	0	2/5	0
M5	0	1/3	1/3	0	1/3

$$\mathbf{AB} = \begin{pmatrix} a & b & c \\ p & q & r \\ u & v & w \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ax + by + cz \\ px + qy + rz \\ ux + vy + wz \end{pmatrix}$$

Movie	User B rating
M1	3
M2	7
M3	8
M4	0 ?
M5	0 ?

Result
4.66
4.54
5.5
3.4
5(recommed)

Implement recommender system with MapReduce

九章算法

- Build co-occurrence matrix
- Build rating matrix
- Multiply co-occurrence matrix and rating matrix

Build Co-occurrence Matrix

What we expect:

User	M1	M2	M3	M4	M5
A	1	1		1	
B	1	1	1		
C		1	1		1
D		1		1	

	M1	M2	M3	M4	M5
M1	2	2	1	1	0
M2	2	4	2	2	1
M3	1	2	2	0	1
M4	1	2	0	2	0
M5	0	1	1	0	1

Build Co-occurrence Matrix

Store raw data in which format?

User	M1	M2	M3	M4	M5
A	1	1		1	
B	1	1	1		
C		1	1		1
D		1		1	

```
1,10001,5.0
1,10002,3.0
1,10003,2.5
2,10001,2.0
2,10002,2.5
2,10003,5.0
2,10004,2.0
3,10001,2.0
3,10004,4.0
3,10005,4.5
3,10007,5.0
4,10001,5.0
4,10003,3.0
4,10004,4.5
4,10006,4.0
5,10001,4.0
5,10002,3.0
5,10003,2.0
5,10004,4.0
5,10005,3.5
5,10006,4.0
```

Build Co-occurrence Matrix

九章算法

Data Preprocessor

Build Co-occurrence Matrix

Divide data by user_id

Build Co-occurrence Matrix

Merge data for same user_id

Divide Data By User

九章算法

Input

user_id1 movie1 rating
user_id2 movie4 rating

user_id34 movie7 rating
user_id49 movie8 rating

Movie3
user_id22 movie4 rating
user_id27 movie7 rating

Output

User_id	Movie_id: Rating	Movie_id: Rating
1	1:10	2: 8
2	1: 8	2: 5.5
3	2: 4.5	8: 9
4	2: 6.5	8: 2.5

Divide Data By User: Mapper

user_id1 movie1 rating
user_id2 movie4 rating

user_id34 movie7 rating
user_id49 movie8 rating

Movie3
user_id22 movie4 rating
user_id27 movie7 rating

User_id	Movie_id	Rating
1	1	10
1	2	8
2	1	8
2	2	5.5
3	2	4.5
3	8	9
4	2	6.5
4	8	2.5

Divide Data By User: Reducer

User_id	Movie_id	Rating
1	1	10
1	2	8
2	1	8
2	2	5.5
3	2	4.5
3	8	9
4	2	6.5
4	8	2.5

Merge →

User_id	Movie_id: Rating	Movie_id: Rating
1	1:10	2: 8
2	1: 8	2: 5.5
3	2: 4.5	8: 9
4	2: 6.5	8: 2.5

Build Co-occurrence Matrix

九章算法

User	M1	M2	M3	M4	M5
A	1	1		1	
B	1	1	1		
C		1	1		1
D		1		1	

	M1	M2	M3	M4	M5
M1	2	2	1	1	0
M2	2	4	2	2	1
M3	1	2	2	0	1
M4	1	2	0	2	0
M5	0	1	1	0	1

Build Co-occurrence Matrix

九章算法

Input

User_id	Movie_id: Rating	Movie_id: Rating
1	1:10	2: 8
2	1: 8	2: 5.5
3	2: 4.5	8: 9
4	2: 6.5	8: 2.5

Output

MovieA: MovieB	Relation
1: 1	2
1: 2	2
2: 1	2
2: 2	4
2: 8	2
8: 2	2
8: 8	2

Build Co-occurrence Matrix

如何计算一部电影被多少人看过？

Build Co-occurrence Matrix

九章算法

User_id	Movie_id: Rating	Movie_id: Rating
1	1:10	2: 8
2	1: 8	2: 5.5
3	2: 4.5	8: 9
4	2: 6.5	8: 2.5

Mapper →

Movie	Count
1	1
2	1
1	1
2	1
2	1
8	1
2	1
8	1

Reducer →

Movie	Sum
1	2
2	4
8	2

Build Co-occurrence Matrix

如何计算两部电影同时被多少人看过？

Build Co-occurrence Matrix: Mapper

九章算法

User_id	Movie_id: Rating	Movie_id: Rating
1	1:10	2: 8
2	1: 8	2: 5.5
3	2: 4.5	8: 9
4	2: 6.5	8: 2.5

MovieA: MovieB	Relation
1:1	1
1:2	1
2:1	1
2:2	1
1:1	1
1:2	1
2:1	1
2:2	1

....

Build Co-occurrence Matrix: Reducer

MovieA: MovieB	Relation
1:1	1
1:2	1
2:1	1
2:2	1
1:1	1
1:2	1
2:1	1
2:2	1

….

Merge →

MovieA: MovieB	Relation
1: 1	2
1: 2	2
2: 1	2
2: 2	4
2: 8	2
8: 2	2
8: 8	2

What you have learned

九章算法

- What is Recommender System
- Methods to implement recommender system
- Understand the theory behind UserCF
- How to implement UserCF in MR