Recommendation system

Jakramate Bootkrajang

September 22, 2016

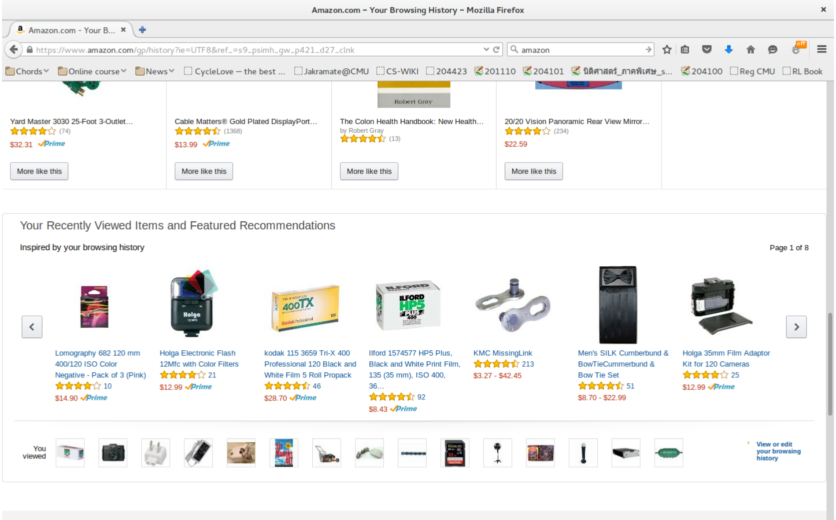
Outlines

- Motivation & Basic Concept
- Content-based system
- Collaborative filtering

Goal

- Predicting user responses to options.
- Applications
  - Suggestions about what customer might like to buy, based on their history of purchases and/or product searches.
  - Offering news articles to on-line readers, based on a prediction of reader interests.
- Examples
  - Netflix, Amazon

Amazon


Amazon.com – Your Browsing History – Mozilla Firefox
Amazon.com - Your B...
https://www.amazon.com/gp/history?ie=UTF8&ref_=s9_psimh_gw_p421_d27_clnk
amazon
Chords
Online course
News
CycleLove – the best ...
Jakramate@CMU
CS-WIKI
204423
201110
204101
204100
Reg CMU
RL Book
Yard Master 3030 25-Foot 3-Outlet...
(74)
$32.31 Prime
More like this
Cable Matters® Gold Plated DisplayPort...
(1368)
$13.99 Prime
More like this
The Colon Health Handbook: New Health...
by Robert Gray
(13)
More like this
20/20 Vision Panoramic Rear View Mirror...
(234)
$22.59
More like this
Your Recently Viewed Items and Featured Recommendations
Inspired by your browsing history
Page 1 of 8
Lomography 682 120 mm 400/120 ISO Color Negative - Pack of 3 (Pink)
10
$14.90 Prime
Holga Electronic Flash 12Mfc with Color Filters
21
$12.99 Prime
kodak 115 3659 Tri-X 400 Professional 120 Black and White Film 5 Roll Propack
46
$28.70 Prime
Ilford 1574577 HP5 Plus, Black and White Print Film, 135 (35 mm), ISO 400, 36...
92
$8.43 Prime
KMC MissingLink
213
$3.27 - $42.45
Men's SILK Cumberbund & BowTieCummerbund & Bow Tie Set
51
$8.70 - $22.99
Holga 35mm Film Adaptor Kit for 120 Cameras
25
$12.99 Prime
You viewed
View or edit your browsing history

Netflix prize

- An open competition for the best collaborative filtering algorithm held by Netflix, an online DVD-rental service.
- Try to predict user ratings for films based on previous ratings.
- The winner must improve the predictive accuracy over Netflix's own system by at least 10%.
- The winner will get the grand prize of US$1,000,0000

Netflix prize winner

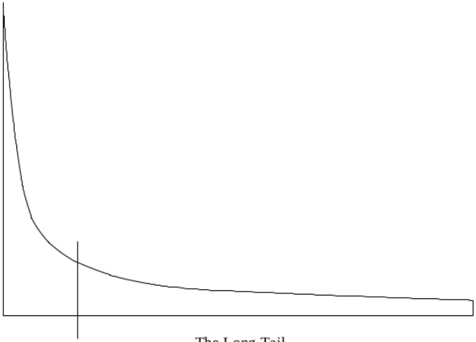
- BellKor's Pragmatic Chaos, a team from AT&T
- It beats Netflix's own algorithm by 10.06%.

Approaches to recommendation systems

1. Content-based systems
   - Examine properties of the items recommended.
   - If a user watched many comedy movies, then recommend a movie classified as comedy.
2. Collaborative filtering
   - Recommends items based on similarity measures between users.
   - Items recommended to a user are those preferred by similar users.
   - Does not use item information.

A long-tail phenomenon

- y-axis => item's popularity, x-axis => item
- Physical shop can only provide what is popular (to the left of the separating line)
- On-line shop can make everything available (whole x-axis)



A utility matrix

- A data for recommendation system are stored as a relationship between *users* and *items* called utility matrix.
- The value in the matrix represents what is known about the degree of preference of that user for that item.
- Matrix is often sparse, meaning most entries are unknown.

A utility matrix representing rating of movies on 1-5 scale

Example 9.1 : In Fig. 9.1 we see an example utility matrix, representing users' ratings of movies on a 1–5 scale, with 5 the highest rating. Blanks represent the situation where the user has not rated the movie. The movie names are HP1, HP2, and HP3 for *Harry Potter* I, II, and III, TW for *Twilight*, and SW1, SW2, and SW3 for *Star Wars* episodes 1, 2, and 3. The users are represented by capital letters A through D.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Figure 9.1: A utility matrix representing ratings of movies on a 1–5 scale

- The goal is to predict the values(rating) for the blank entries.

Filling the utility matrix

- Direct approach
  - Ask users to rate items.
  - However, most users are not willing to do so.
  - Score can be 1 to N.
- Indirect approach
  - Making inference.
  - If user watched the movie then the user can be said to 'like' this item.
  - Or if user viewed the item we can assume that he likes the item even if he did not make a purchase.
  - Score can be either 0 or 1.

Content-based Recommendation

- Focuses on properties of items.
- Similarity of items is determined by measuring the similarity in their properties.
- Recall some similarity measures
  - Cosine similarity
  - Minkowski distance

Item profiles

- To represent an item in the system we must construct for each item a *profile*, (or features)
- Profile consists of some characteristics of the item.
- For example, a movie could be represented by
  - The set of actors
  - The director
  - The year
  - The genre
- These set of features are readily available.
- How about books?, text documents? or images ?

User profiles

- Contains the same features as item profiles that describe the user's preferences.
- Can be estimated using information from item profiles and utility matrix

Examples

- Suppose items are movies, represented by boolean features corresponding to actors.
- Item profiles

Movie / Actor	S.Johansson	C.Evans	R.Downey
Ant man		1	
Avenger	1	1	1
Iron man	1		1

- Utility matrix

User / Movie	Ant man	Avenger	Iron man
Suchart	1		
Je			1
Yoon	1		1

Which movie should we recommend to Je?

- Je's profile (originally empty)

User / Actor	S.Johansson	C.Evans	R.Downey
Je	0	0	0

- The preference for component j of user i can be estimated by
  $p_{i,j} = \sum_{S_i}(\delta(j \in k))/|S_i|$, where
  S_i is a set of items voted by user i.
  $\delta(j \in k)$ is 1 if item k contain feature j, and 0 otherwise.

Which movie should we recommend to Je? (cont.)

- So, one out of 1 movie that Je likes (watched) has S.Johansson as one of the actors. Her profiles then has $1/1 = 1$ in the component for S.Johansson.
- Je's new profile

User / Actor	S.Johansson	C.Evans	R.Downey
Je	1	0	1

- Based on cosine similarity, Je's preference is more similar to Avenger than Ant man, so we should recommend Avenger to Je.
- Je's user profile will regularly be updated as more information becomes available.

What if the utility matrix stores rating score?

- Score-based utility matrix (1-5)

User / Movie	Ant man	Avenger	Iron man
Suchart	3		
Je			2
Yoon	5		5

- The preference for component j of user i can then be estimated by

  $p_{i,j} = \sum_{c \in C_j} (v_c - \bar{v}_i)/|C_{i,j}|$, where

  $C_{i,j}$ is a set of items voted by user i which has j as its feature.

  $\bar{v}_i$ is average score given by user i

Collaborative Filtering

- Another different approach to recommendation systems.
- Instead of using features of items to determine similarity
- Collaborative filtering recommends items based on similarity of users.
- Two steps of collaborative filtering
  - Identifying similar users.
  - Recommending what similar users like.

Algorithm for Collaborative filtering

- The simplest is memory-based algorithm
- Given
  - $v_{i,j}$ = vote of user i on item j.
  - I_i = items for which user i has voted.
  - Mean vote of i is $\bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}$
- Predicted vote for "active user" a is weighted sum

  $p_{a,j} = \bar{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \bar{v}_i)$

  where κ is a normaliser

  and $w(a,i)$ is the weight given by the similarity between user a and user i

Measuring similarity between users

- How ?

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

- Some important similarity measures
  - Nearest neighbour based on Minkowski distance
  - Pearson correlation coefficient
  - Jaccard distance
  - Cosine distance

Nearest neighbour

- K-nearest users

$$w(a,i) = \begin{cases} 1, & \text{if } \; i \in \text{neighbour(a)} \\ 0, & \text{otherwise} \end{cases}$$

- where neighbour(a) is a set containing k users with minimum distance from user a
- Distance is measured by Minkowski distance
- $d(u_i, u_j) = (|u_{i1} - u_{j1}|^h + \cdots + |u_{iM} - u_{jM}|^h)^{\frac{1}{h}}$
- Another variant: using $d(u_i, u_j)$ in place of 1.

Pearson correlation coefficient

- Defined as:

$$w(i,j) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}}$$

- Represents correlation of voting scores between user a and user i
- Positive value indicates user a tends to give the same score as user i.
- Negative value says the opposite.

Jaccard distance

- The Jaccard similarity of sets S and T is $|S \cap T|/|S \cup T|$
- That is, the ratio of the size of the intersection of S and T to the size of their union.
- Example

  S = {dog, cat, parrot, monkey}

  T = {dog, monkey, snake}

  Then $SIM_{Jaccard}(S, T) = 2/5 = 0.4$
- Jaccard distance is 1 - $SIM_{Jaccard}$

Jaccard distance (cont.)

- In the context of collaborative filtering the utility matrix must be converted to binary, i.e., is the item rated or not rated?
- From our utility matrix, what's the Jaccard similarity between user A and B ?

Cosine distance

- $SIM_{cos}(A, B) = \frac{A \cdot B}{||A|| \cdot ||B||}$
  where A is a rating vector of user *a* and B is a rating vector of user *b*.
- In the context of collaborative filtering the unrated entries in the utility matrix could be treat as a 0 value.
- From our utility matrix, what's the cosine similarity between user A and B?

Rounding data

- Sometimes we may want to perform some discretisation on the data.
- This simplifies the calculation of distance and eliminates apparent similarity between movies a user rates highly and those with low scores. (especially for Jaccard similarity measure)
- We could take 3,4,5 as '1' and consider ratings 1 and 2 as unrated.
- What is the Jaccard similarity of user A and B after this modification then?

Normalising ratings

- We can subtract from each rating the average rating of that user.
- From this, we turn low rating into negative numbers and high ratings into positive numbers.
- This method is normally suitable for pairing with cosine similarity measure.
- What is the cosine similarity of user A and B after this modification then?

How to evaluate our system?

- Hold out some of users rating for testing purpose
- Performance can be base on accuracy (if our utility matrix stores binary values)
- Performance can be based on mean square error (if our utility matrix scores actual rating values)
- $M.S.E = \frac{\sum_{i=1}^{N}(p_{i,j} - vi,j)^2}{N}$

References

- Mining From Massive Data: Anand Rajaraman, Jure Leskovec, Jeffrey D. Ullman
- Jannach, Dietmar, and Gerhard Friedrich. "Tutorial: Recommender Systems." Proceedings of the International Joint Conference on Artificial Intelligence, Barcelona. 2011.