CSCI4622 Machine Learning

2020 Spring

Unsupervised Learning

Geena Kim

Recommender System-continued

- Content-based
- Collaborative Filtering
  - Similarity
  - Matrix Factorization

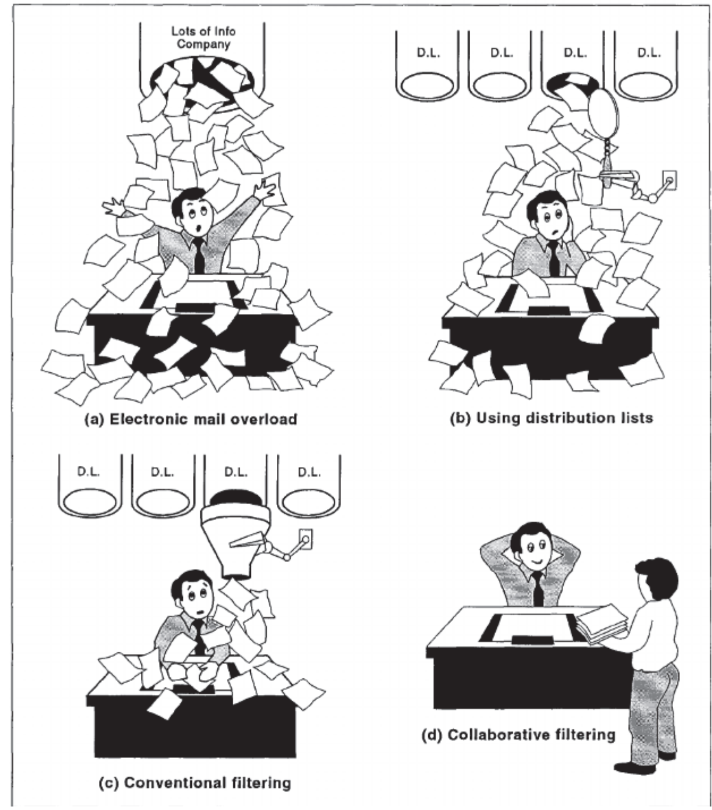
Collaborative Filtering

- No need of hand-engineered features
- Domain-free
- Can learn latent info that are hard to profile using content filtering
- Generally more accurate about user preference
- May suffer from “cold-start” problem

C

Tapstry

The origin of the name: filtering documents in emails



Goldberg et. al., (1992)

Collaborative Filtering

Collaborative filtering:

- Only consider past user behavior.

(**not** content properties...)

- User-User similarity
- Item-Item similarity

Similarity-based Collaborative Filtering:

- Use similarity metric between users or items
- Use the similarity matrix directly or with clustering

Matrix Factorization Methods:

- Find latent features/factors
- Still needs the similarity matrix

Similarity

n 600M

Item

User	A	B	C	D	...
Al	1	?	2	?	
Bob	?	2	3	4	
Cat	3	?	1	5	
Dan	?	2	?	?	
Ed	2	?	?	1	
...					

100M

m

$mC_2 \sim O(m^2)$

User-user similarity : $O(m^2 \times n)$

Item-item similarity : $O(n^2 m)$

Calculation cost

Let:
m = #users,
n = #items

User-User: $O(m^2n)$
Item-Item: $O(mn^2)$

Similarity Measure

0 ~ 1

Distance-based

0 ~ inf

- Manhattan distance c=1
- Euclidean distance c=2
- Minkowski distance

$\left(\sum_i (x_A^i - x_B^i)^c\right)^{1/c}$

$$\text{similarity}(a,b) = \frac{1}{1 + \text{dist}(a,b)}$$

1

0

$+\infty$

Similarity Measure

a

b

Pearson Correlation

$$\frac{\mathrm{cov}(a,b)}{\mathrm{std}(a) * \mathrm{std}(b)} = \frac{\sum_i (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_i (a_i - \bar{a})^2}\sqrt{\sum_i (b_i - \bar{b})^2}}$$

$-1 \sim 1$

$$\mathrm{similarity}(a,b) = 0.5 + 0.5 * \mathrm{pearson}(a,b)$$

$0 \sim 1$

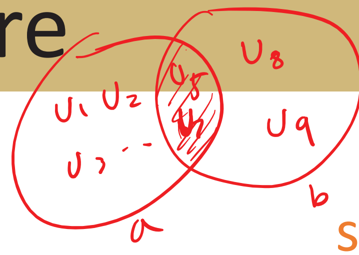
Similarity Measure

√ Cosine similarity

$$\frac{a \cdot b}{||a||||b||} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2}\sqrt{\sum_i b_i^2}}$$ -1 ~ 1

$$\text{similarity}(a, b) = 0.5 + 0.5 * \cos(\theta_{a,b})$$

Similarity Measure

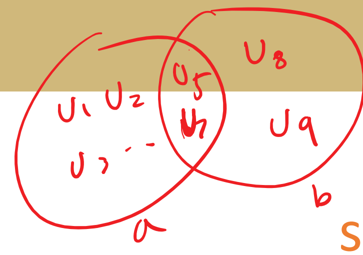


Jaccard similarity

set of users who rated item a

$$\text{similarity}(a,b) = \frac{|U_a \cap U_b|}{|U_a \cup U_b|}$$

Similarity



Jaccard similarity

set of users who rated item a

$$\text{similarity}(a,b) = \frac{|U_a \cap U_b|}{|U_a \cup U_b|}$$

Choosing a similarity metric

How do we choose which metric to use?

Example: Movie ratings by users

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

Intuitively, what info should a metric capture?

What should we do about missing values?

Choosing a similarity metric

Example: Movie ratings by users

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4 −3.3			5 −3.3	1 −3.3	3.3	
B	5 − 4.x	5 − 4.x	4 − 4.x				
C				2 −	4 −	5	
D		3 −3 0					3 −3 0

$$\frac{\vec{a} \cdot \vec{b}}{|a|\,|b|}$$

(e.g.) Jaccard

(e.g.) Cosine

0

−1 (opposite)

1 (close)

Choosing a similarity metric

If we want to use Jaccard... we could

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

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	1			1	0		
B	1	1	1				
C				0	1	1	
D		1					1

What can we capture and cannot capture?

Choosing a similarity metric

What about Cosine similarity?

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

Cosine is -1~1

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	$2/3$			$5/3$	$-7/3$		
B	$1/3$	$1/3$	$-2/3$				
C				$-5/3$	$1/3$	$4/3$	
D		0					0

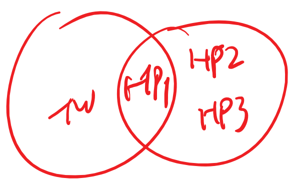
Now, it can capture the opposite preferences

Choosing a similarity metric

Jaccard similarity calculation example

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	1			1			
B	1	1	1				
C					1	1	
D		1					1

What to do with missing values?

Jaccard(A,B) 

$\rightarrow$ 1/4 ?

Jaccard(A,C) 0

Choosing a similarity metric

Cosine similarity calculation example

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	$2/3$			$5/3$	$-7/3$		
B	$1/3$	$1/3$	$-2/3$				
C				$-5/3$	$1/3$	$4/3$	
D		0					0

Cos(A,B)

$$\frac{(2/3)\times(1/3)}{\sqrt{(2/3)^2+(5/3)^2+(-7/3)^2}\sqrt{(1/3)^2+(1/3)^2+(-2/3)^2}} = 0.092$$

Cos(A,C)

$$\frac{(5/3)\times(-5/3)+(-7/3)\times(1/3)}{\sqrt{(2/3)^2+(5/3)^2+(-7/3)^2}\sqrt{(-5/3)^2+(1/3)^2+(4/3)^2}} = -0.559$$

Clustering the utility matrix

How do we deal with sparcity?

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

	HP	TW	SW
A	4	5	1
B	4.67		
C		2	4.5
D	3		3

Similarity Matrix

Raw utility matrix (e.g. movie ratings table)

User \ Item	A	B	C	D	...
Al	1	?	2	?	
Bob	?	2	3	4	
Cat	3	?	1	5	
Dan	?	2	?	?	
Ed	2	?	?	1	
...					

Similarity matrix (e.g. item-item)

	item 1	item 2	item 3	...
item 1	1	0.3	0.2	...
item 2	0.3	1	0.7	...
item 3	0.2	0.7	1	...
...	...	...	...	...

Predicting missing values

Item $r_{u,j}$

User

	A	B	C	D	...
Al	1	?	2	?	
Bob	?	2	3	4	
Cat	3	?	1	5	
Dan	?	2	?	?	
Ed	2	?	?	1	
...					

	item 1	item 2	item 3	...
item 1	1	0.3	0.2	...
item 2	0.3	1	0.7	...
item 3	0.2	0.7	1	...
...	...	...	...	...

Using Neighborhood method

$$\text{rating}(u,i) = \frac{\sum_{j \in I_u} \text{similarity}(i,j) * r_{u,j}}{\sum_{j \in I_u} \text{similarity}(i,j)}$$

$$I_u = \text{ set of items rated by user } u$$
$$r_{u,j} = \text{ user } u\text{'s rating of item } j$$

$$\text{rating}(u,i) = \frac{\sum_{j \in I_u \cap N_i} \text{similarity}(i,j) * r_{u,j}}{\sum_{j \in I_u \cap N_i} \text{similarity}(i,j)}$$

$$I_u = \text{ set of items rated by user } u$$
$$r_{u,j} = \text{ user } u\text{'s rating of item } j$$
$$N_i \text{ is the } n \text{ items which are most similar to item } i$$

Predicting missing values

	Item				
User	**A**	**B**	**C**	**D**	**...**
Al	1	?	2	?	
Bob	?	2	3	4	
Cat	3	?	1	5	
Dan	?	2	?	?	
Ed	2	?	?	1	
...					

n

m

Using Latent factor model

k

$$R_{m \times n} \approx U_{m \times k} V_{k \times n}$$

k:
# of latent features
is a hyperparameter

Update U(i,k)

$$U'(i,k) = x = \frac{\sum_j v_{kj}\left(m_{ij} - \sum_{k' \neq k} u_{ik'} v_{k'j}\right)}{\sum_j v_{kj}^2}$$

Update V(k,j)

$$V'(k,j) = y = \frac{\sum_i u_{ik}\left(m_{ij} - \sum_{k' \neq k} u_{ik'} v_{k'j}\right)}{\sum_i u_{ik}^2}$$

We can update more slowly

α or γ (step size or learning rate)

$$\Delta U(i,k) = U'(i,k) - U(i,k)$$

$$\Delta V(k,j) = V'(k,j) - V(k,j)$$

Using the smaller learning rate:

$$U(i,k) \leftarrow U(i,k) + \gamma \Delta U(i,k)$$

$$V(k,j) \leftarrow V(k,j) + \gamma \Delta V(k,j)$$

References

[1] Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the Association of Computing Machinery, 35*(12), 61–70.

[2] Leskovec, J., Rajaraman, A., & Ullman, J. D., Mining Massive Datasets Ch.9 Recommender Systems http://infolab.stanford.edu/~ullman/mmds/ch9.pdf