Introduction to Recommendation System

Content

Overview of Recsys

Why needs recsys?

Applications of recsys

Recommendation approaches

Recommendation evaluations

Why needs recsys?

- Too much information
- The paradox of choice
- Traditional recommendation
  - Ask friends
  - Ask sellers

Why needs recsys?

- Recsys vs Information retrieval
  - Information retrieval: User need is provided
  - Recsys: Users don't know what they want

Why needs recsys?

- Customers vs sellers
- Sellers:
  - Customer satisfaction
  - Keep customers
  - Sell goods
- Customers:
  - Show me things like what I've purchased
  - Show me things people like me purchase

Applications of recsys

- E-commerce
  - Hi-tech
  - Fashion
  - Foods
  - etc

Applications of recsys

- Entertainments
  - Movies
  - Photo
  - Books
  - Online game
- News
- Question-answering

- Research activities
  - Article recommendation
- Service
  - Tourism
  - Restaurants, hotels, flights
- Educations
  - E-learning
- Software applications
- Sourcecode

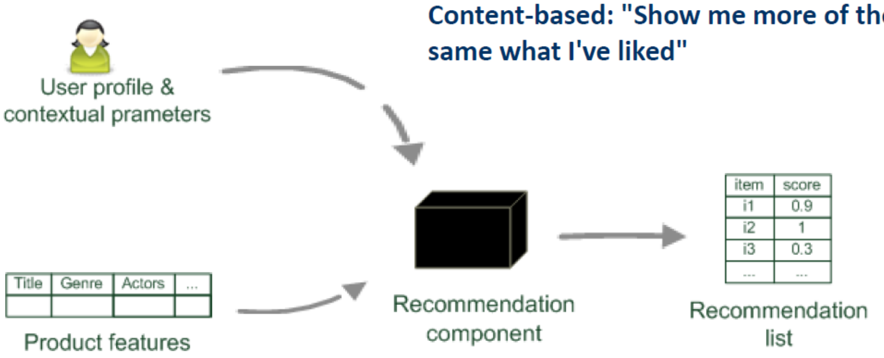
- User-user recommendation
  - Research collaboration
  - Friend recommendation
  - Dating

- Amazon: increases 30% revenue
- Netflix: increases $1B per year
- Google news: increases 40% traffic

- Content-based recommendation
- Collaborative filtering
- Session-based recommendation
- Hybrid approach

Introduction to Recsys
Overview of recsys
Recommendation approaches

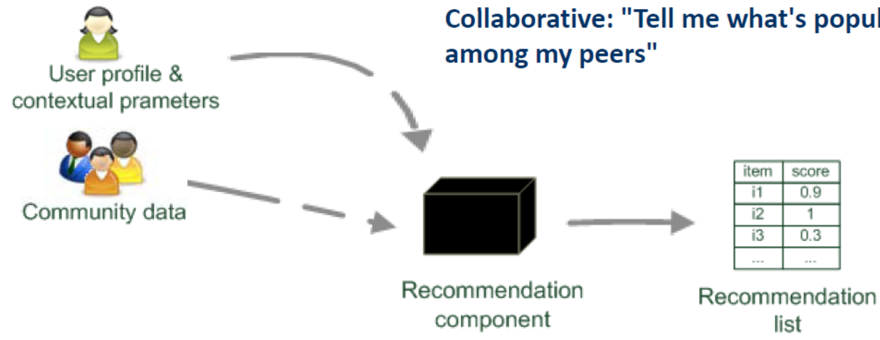
Content-based: "Show me more of the same what I've liked"



from Recommender Systems: An Introduction

Introduction to Recsys
Overview of recsys
Recommendation approaches

Collaborative: "Tell me what's popular among my peers"



from Recommender Systems: An Introduction

Introduction to Recsys
Overview of recsys
Recommendation approaches

- Collaborative filtering (CF):
  - KNN
  - Matrix factorization
  - Latent models

Introduction to Recsys
Overview of recsys
Recommendation approaches

- Main challenges
  - Sparse data
  - Cold-start
  - Stream data
  - Scale-up

- Problem statement: Predicting rating of user *u* for item *i*: $r_{u,i}$
- Offline evaluation
- Online evaluation

- Offline evaluation
  - Rating: Likert scale
    - 1: very bad
    - 2: bad
    - 3: neutral
    - 4: good
    - 5: very good
  - Binary
    - Based on likert scale
    - Based on implicit feedbacks (click, view, like/dislike, comment, purchase)

- Sử dụng dữ liệu lịch sử của người dùng

  (người dùng, sản phẩm, đánh giá)
- Chia thành dữ liệu train/dev/test
- So sánh dữ liệu kết quả của hệ thống với dữ liệu test
  - $r_{i,j}$: Đánh giá thực sự của người dùng *i* với sản phẩm *j*
  - $p_{i,j}$: Dự đoán của hệ thống

- Mean Absolute Error

$$MAE = \frac{\sum_{\{i,j\}} \left| p_{i,j} - r_{i,j} \right|}{n},$$

- Normalized Mean Absolute Error

$$NMAE = \frac{MAE}{r_{\max} - r_{\min}},$$

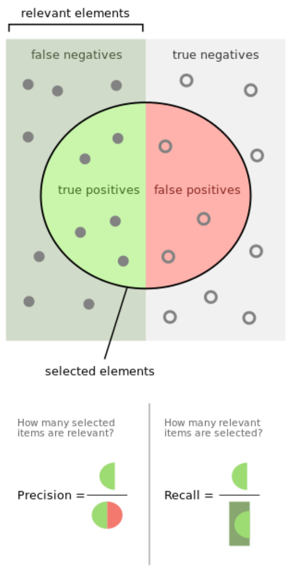
- Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{\{i,j\}}\left(p_{i,j} - r_{i,j}\right)^2},$$

- Precision/Recall/F-score

Actual	Predicted	
	Positive	Negative
Positive	TruePositive	FalseNegative
Negative	FalsePositive	TureNegative

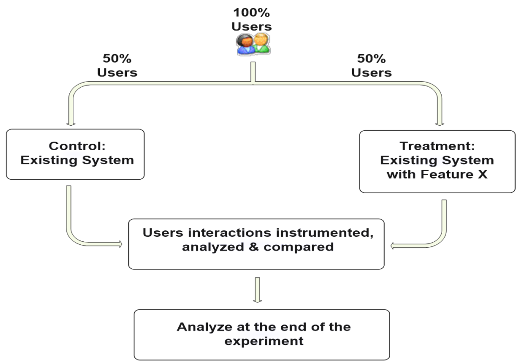
- Precision = TP / (TP + FP)
- Recall = TP / (TP + FN)
- F = 2PR / (P + R)



from Wikipedia

Recommendation evaluations

- Online evaluation: A/B Testing

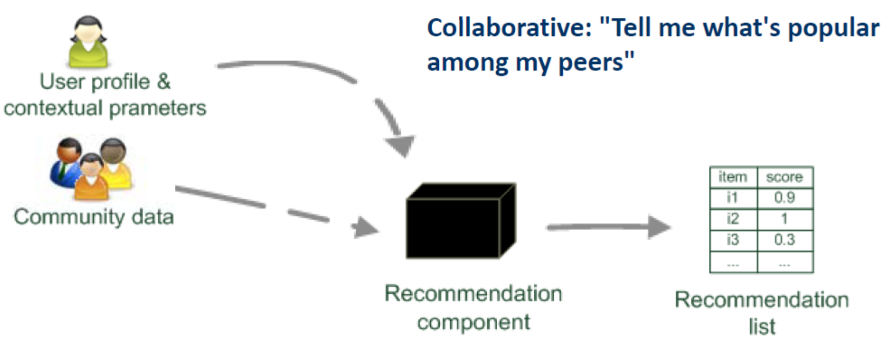


from Online Controlled Experiments...

Recommendation evaluations

- Recommendation based on similar users
- Approaches
  - K-nearest neighbors (KNN)
  - Matrix factorization
  - Latent models

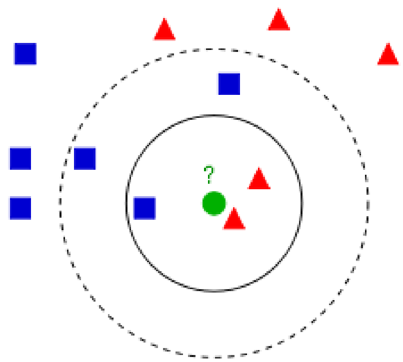
Collaborative filtering



from Recommender Systems: An Introduction

KNN

- A machine learning method for
  - Classification
  - Regression
- Non-parametric, memory-based
- Hyper-param
  - K
  - Similarity (distance) metric



from Wikipedia

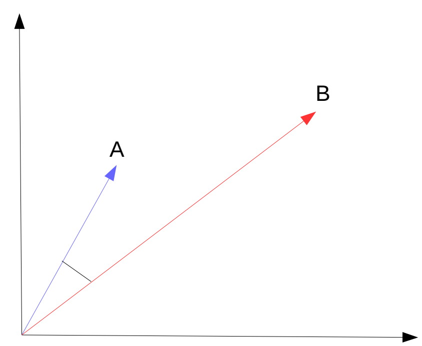
- Given:
  - An *item* vector
  - A similarity matric between a pair of items *sim(item$_i$,item$_j$)*

B1. Find k nearest neighbor items

B2. Find class/regression values

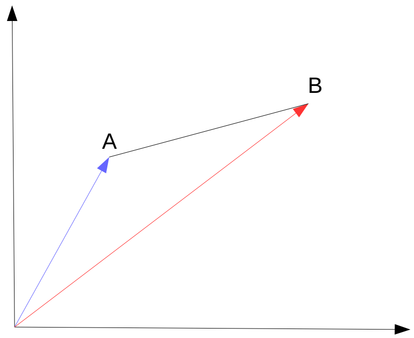
- Cosine
- Euclide
- Manhattan
- Jaccard
- Pearson

- cosine(A,B) = AB / ($||A||_2$ $||B||_2$)



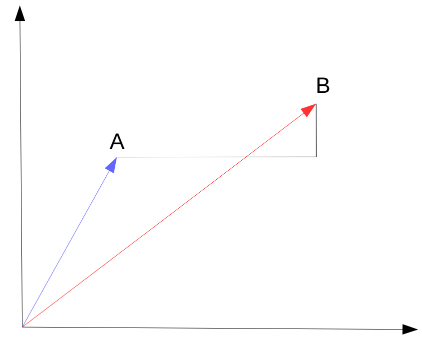
Introduction to recsys
KNN-based collaborative filtering
KNN

- Euclide(A,B) = $||A - B||_2$



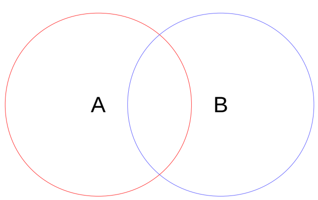
Introduction to recsys
KNN-based collaborative filtering
KNN

- Manhattan(AB) = $||A - B||_1$



Introduction to recsys
KNN-based collaborative filtering
KNN

- Jaccard(A,B) = A ∩B / A ∪ B



Introduction to recsys
KNN-based collaborative filtering
KNN

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Pearson correlation:
  - X, Y are random variables
  - *n*: number of samples
  - x_i: i^{th} sample

Introduction to recsys
User-based CF

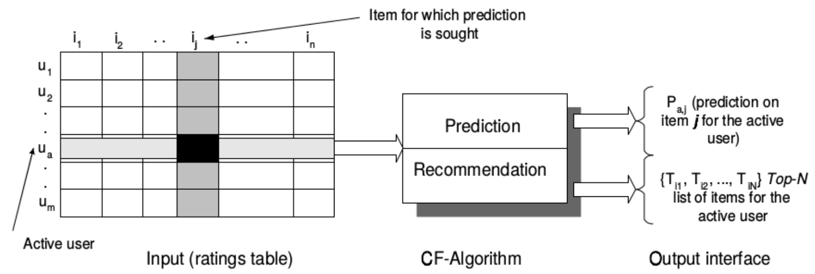
- Utility matrix contains user-item ratings $r_{u,i}$
- Predict rating of user u for item i: $p_{u,i}$
- User-based CF:

B0. Calculate u vector

B1. Find k nearest users

B2. Calculate $p_{u,i}$

Introduction to recsys
User-based CF
Utility matrix



from B. Sarwar et al 2010

Introduction to recsys
User-based CF
Pearson correlation

$$sim(\mathbf{u},\mathbf{v}) = \frac{\sum_{i \in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sqrt{\sum_{i \in C}(r_{\mathbf{u},i} - \bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{i \in C}(r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})^2}},$$

u: user

C: Set of common items that u and v both rated

$r_{\boldsymbol{u},i}$: rating of user u for item i

$\bar{r}_{\boldsymbol{u}}$: Averaged rating of u

Introduction to recsys
User-based CF
Rating prediction

$$p(\mathbf{u},i) = \bar{r}_{\mathbf{u}} + \frac{\sum_{\mathbf{v} \in V} sim(\mathbf{u},\mathbf{v}) \times (r_{\mathbf{v},i} - \bar{r}_{\mathbf{v}})}{\sum_{\mathbf{v} \in V} |sim(\mathbf{u},\mathbf{v})|},$$

V: k-nearest neighbors of user u

Example (user-based, k = 2)

	item_1	item_2	item_3	item_4
user_1	5	4	4	1
user_2	2	1		
user_3	5	4	4	?
user_4		1	2	5

$\bar{r}_1 = 14/4$ $\quad sim(u_3,u_1) = \sim 0.492$ $\quad p(u_3,i_4) = \sim 5.481$

$\bar{r}_2 = 3/2$ $\quad sim(u_3,u_2) = \sim 0.948$

$\bar{r}_3 = 13/3$ $\quad sim(u_3,u_4) = \sim 0.919$

$\bar{r}_4 = 8/3$

Shortcomings

- Sparse data
  - VD: The most active user in Amazon just rates 1% of books (1% of 2M books is 20,000)
- Usually re-calculates user vectors
- Scaling-up: Calculate over all users

Item-based CF

- Utility matrix
- Predict $p_{u,i}$
- Item-based recommendation

  B0. Represent item vectors

  B1. Find k-nearest neighbor items

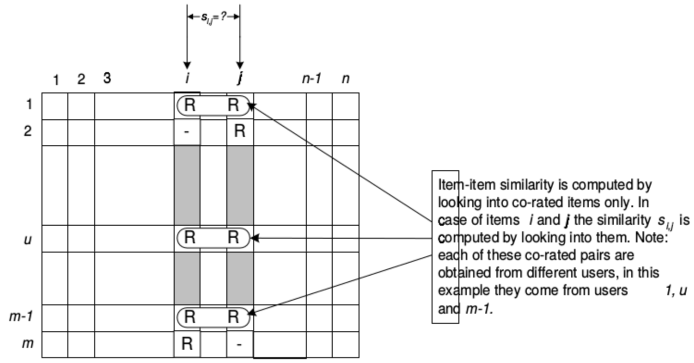
  B2. Calculate $p_{u,i}$

Item-based CF

- Suitable for system with #users >> #items
- Dense item vectors
- Calculate offline item-item similarity

Introduction to recsys

Item-based CF



from B. Sarwar et al 2010

Introduction to recsys

Item-based CF

$$sim(i,j) = \frac{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},i}-\bar{r}_{\mathbf{u}})(r_{\mathbf{u},j}-\bar{r}_{\mathbf{u}})}{\sqrt{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},i}-\bar{r}_{\mathbf{u}})^2}\sqrt{\sum_{\mathbf{u}\in U}(r_{\mathbf{u},j}-\bar{r}_{\mathbf{u}})^2}},$$

i, j: item

U: set of common users both rated *i* and *j*

Introduction to recsys

Item-based CF

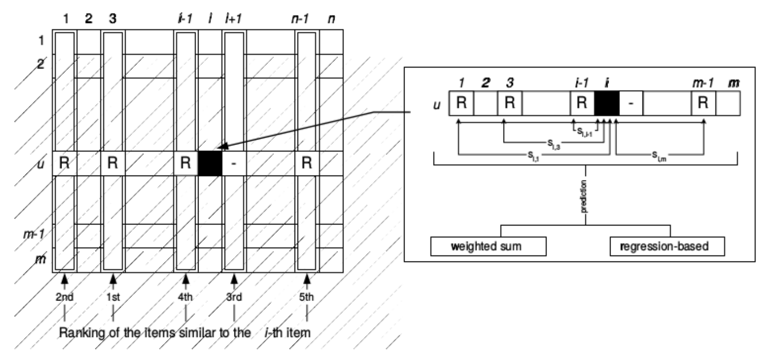
$$p(\mathbf{u},i) = \frac{\sum_{j\in J} r_{\mathbf{u},j} \times sim(i,j)}{\sum_{j\in J} sim(i,j)},$$

J: k nearest neighbor items of *i*

Introduction to recsys

Item-based CF

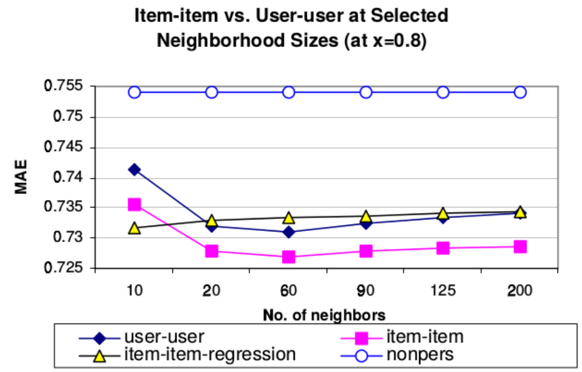
Example

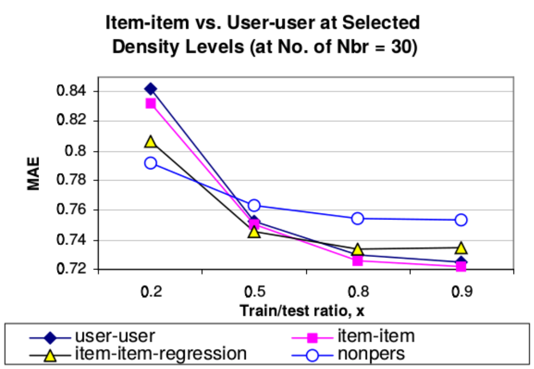


from B. Sarwar et al 2010

- Mean Absolute Error

$$MAE = \frac{\sum_{\{i,j\}} \left| p_{i,j} - r_{i,j} \right|}{n},$$





Q&A

Mail to: hieunk@soict.hust.edu.vn