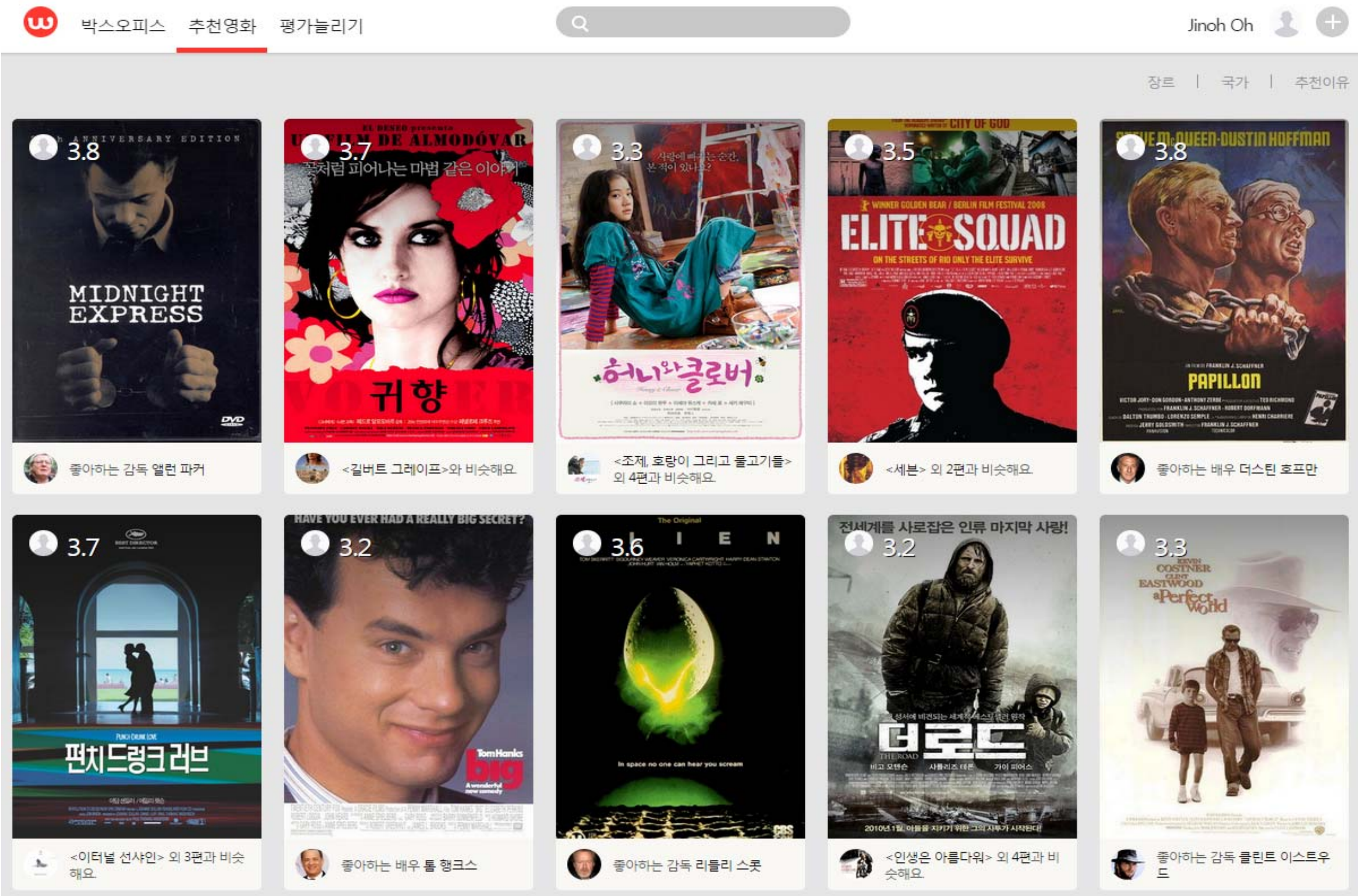
Week 12-1

Recommender System I

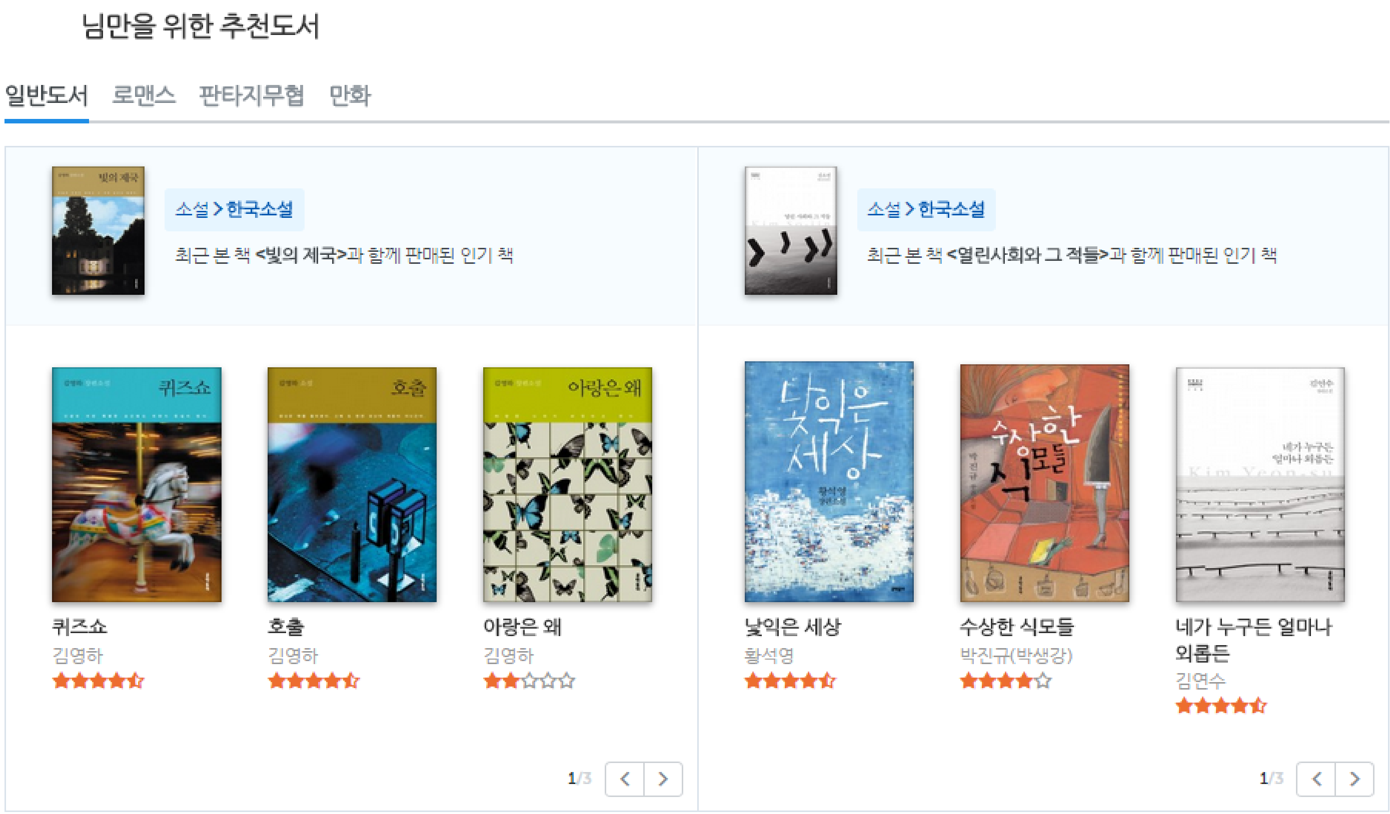
Big Data

Prof. Hwanjo Yu

Recommendation



Recommendation



Other examples

- Movie recommendation (Netflix)
- Related product recommendation (Amazon)
- Web page ranking (Google)
- Social recommendation (Facebook)
- News content recommendation (Yahoo)
- Priority inbox & spam filtering (Google)
- Online dating (OK Cupid)
- Computational Advertising (Yahoo)

The value of recommendation

- Netflix: 2/3 of the movies watched are recommended
- Google News: recommendations generate 38% more clickthrough
- Amazon: 35% sales from recommendations
- ChoiceStream: 28% of the people would buy more music if they found what they liked

Traditional problem statement

- Goal: Predict the rating of users on unseen items
- Measure: Root Mean Square Error (RMSE)
  - $RMSE(S) = \sqrt{\frac{1}{|S|}\sum_{(i,j)\in S}\left(r_{ij} - \widehat{r_{ij}}\right)^2}$
- How? – find a function $\mathrm{r_{ij}} \approx \widehat{r_{ij}} = f(\text{user's history, unseen item})$

An unseen Item

Users' history

Conventional approaches for recommendation

- Memory-based recommendation
  - K-nearest neighbor
- Model-based recommendation
  - Matrix-factorization based recommendation

K-NN Memory-Based Recommendation

Big Data

Key idea of K-Nearest-Neighbor (KNN)

- If two people A and B have the same preference on an product X and B prefers another product Y, then A is likely to prefer Y too.
- Two procedure
  - Find people with similar preference
  - Exploits other's experience when choosing products

Following questions

- How to detect people having similar preference?
  - How to ***represent*** each individuals (or products)
  - How to define ***similarity*** between individuals (or products)
- Solution
  - Model users and items as vectors
  - Use similarity measure for vectors
    - Inner product
    - Cosine similarity
    - Pearson correlation

Comparison on two ways generating vectors

Content-based approach (CB)

- Domain specific
  - Movie domain – actor, genre, director, year, description words
  - Music domain – singer, genre, composer, lyrics

Collaborative Filtering (CF)

- Domain independent
  - A product is identified by a set of users who purchased the item
  - A user is identified by a set of products which the user purchased
- Rating information turns out to produce more *"accurate"* results

K-nearest neighbor

- User-based nearest-neighbor collaborative filtering [Resnick 94]

	Item 1	Item 2	Item 3	Item 4	Item 5
User A	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

K-nearest neighbor

- User-based nearest-neighbor collaborative filtering [Resnick 94]

	Item 1	Item 2	Item 3	Item 4	Item 5
User A	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

- Pearson correlation

$$sim(u_1, u_2) = \frac{\sum_{i \in I^{\{1,2\}}} (r_{1i} - \overline{r_1})(r_{2i} - \overline{r_2})}{\sqrt{\sum_{i \in I^{\{1,2\}}} (r_{1i} - \overline{r_1})^2} \sqrt{\sum_{i \in I^{\{1,2\}}} (r_{2i} - \overline{r_2})^2}}$$

$I^{\{x,y\}}$: A set of items, rated by both user x and user y
r_{ij} : a rating on item j by user i
$\overline{r_i}$: an average rating of user i

Modified from Dietmar Jannach, Gerhard Fridrich, "Tutorial: Recommender system s", IJCAI 2013

K-nearest neighbor

- User-based nearest-neighbor collaborative filtering [Resnick 94]

	Item 1	Item 2	Item 3	Item 4	Item 5
User A	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

$$sim(u_1, u_2) = \frac{\sum_{i \in I^{\{1.2\}}} (r_{1i} - \overline{r_1})(r_{2i} - \overline{r_2}\ )}{\sqrt{\sum_{i \in I^{\{1,2\}}} (r_{1i} - \overline{r_1})^2} \sqrt{\sum_{i \in I^{\{1,2\}}} (r_{2i} - \overline{r_2})^2}}$$

$$\overline{r_A} = \frac{5+3+4+4}{4} = 4$$

$$\overline{r_1} = \frac{3+1+2+3+3}{5} = 2.4$$

$$\overline{r_2} = \frac{4+3+4+3+5}{5} = 3.8$$

$$\overline{r_3} = \frac{3+3+1+5+3}{5} = 3.2$$

$$\overline{r_4} = \frac{1+5+5+2+1}{5} = 2.8$$

$$sim(u_A, u_1) = \frac{(5-4)(3-2.4) + (3-4)(1-2.4) + (4-4)(2-2.4) + (4-4)(3-2.4)}{\sqrt{(5-4)^2 + (3-4)^2 + \cdots} \sqrt{(3-2.4)^2 + (1-2.4)^2 + (2-2.4)^2 + (3-2.4)^2}}$$

$$\approx 0.84$$

Modified from Dietmar Jannach, Gerhard Fridrich, “Tutorial: Recommender system s”, IJCAI 2013

K-nearest neighbor

- User-based nearest-neighbor collaborative filtering [Resnick 94]

	Item 1	Item 2	Item 3	Item 4	Item 5
User A	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

$$sim(u_1, u_2) = \frac{\sum_{i \in I^{\{1.2\}}} (r_{1i} - \overline{r_1})(r_{2i} - \overline{r_2})}{\sqrt{\sum_{i \in I^{\{1,2\}}} (r_{1i} - \overline{r_1})^2} \sqrt{\sum_{i \in I^{\{1,2\}}} (r_{2i} - \overline{r_2})^2}}$$

1-NN, sim(UserA, user 1) = 0.84

2-NN, sim(UserA, user 2) = 0.42

- Final prediction

$$\widehat{r_{ui}} = \overline{r_u} + \frac{\sum_{k \in N} sim(u,k) * (r_{ki} - \overline{r_k})}{\sum_{k \in N} sim(u,k)},$$ where N is a K-nearest neighbor set

Modified from Dietmar Jannach, Gerhard Fridrich, “Tutorial: Recommender system s”, IJCAI 2013

K-nearest neighbor

- User-based nearest-neighbor collaborative filtering [Resnick 94]

	Item 1	Item 2	Item 3	Item 4	Item 5
User A	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

$$\widehat{r_{ui}} = \overline{r_u} + \frac{\sum_{k \in N} sim(u,k) * (r_{ki} - \overline{r_k})}{\sum_{k \in N} sim(u,k)}$$

1-NN, sim(UserA, user 1) = 0.84
2-NN, sim(UserA, user 2) = 0.42

- Final prediction

$$\widehat{\mathrm{r}_{A5}} = \overline{r_A} + \frac{0.84 \cdot (3 - 2.4) + 0.42 \cdot (5 - 3.8)}{0.84 + 0.42} = 4.88$$

Modified from Dietmar Jannach, Gerhard Fridrich, “Tutorial: Recommender system s”, IJCAI 2013

K-nearest neighbor

- Item-based collaborative filtering recommendation algorithm [Sarwar 2001]

	Item 1	Item 2	Item 3	Item 4	Item 5
User A	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

Modified from Dietmar Jannach, Gerhard Fridrich, “Tutorial: Recommender system s”, IJCAI 2013

K-nearest neighbor

- Item-based collaborative filtering recommendation algorithm [Sarwar 2001]

	Item 1	Item 2	Item 3	Item 4	Item 5
User A	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

- Find KNN Items that are similar to Item 5
- Cosine similarity

$$sim(I_i, I_j) = \frac{I_i \cdot I_j}{|I_i||I_j|}$$

Modified from Dietmar Jannach, Gerhard Fridrich, "Tutorial: Recommender system s", IJCAI 2013

K-nearest neighbor

- Item-based collaborative filtering recommendation algorithm [Sarwar 2001]

	Item 1	Item 2	Item 3	Item 4	Item 5
User A	5	3	4	4	?
User 1	3	1	2	3	3
User 2	4	3	4	3	5
User 3	3	3	1	5	4
User 4	1	5	5	2	1

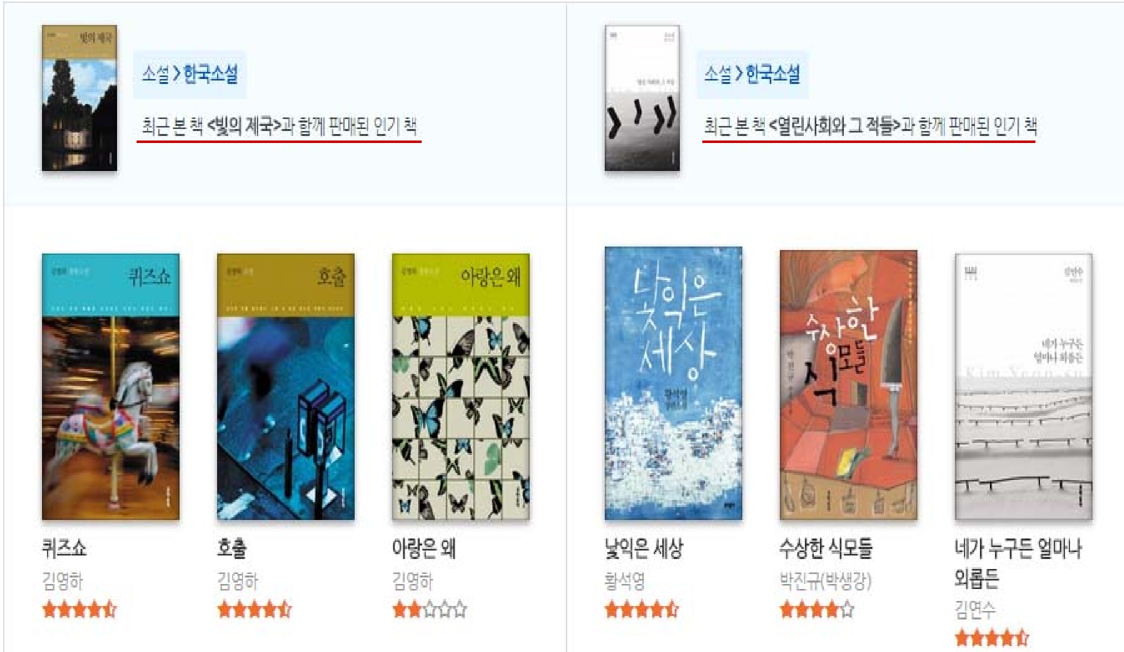
- Prediction
  - Take User’s ratings for these items to predict the rating for the Item 5

K-nearest neighbor

- User-based collaborative filtering
  - K-nearest **users** using user-user similarity
  - Predict the final score by user-item similarity of neighbors
- Item-based collaborative filtering
  - K-nearest **items** using item-item similarity
  - Predict the final score by user-item similarities for the neighbor items

Properties

- Intuitive
- No (substantial) training
- Easy to explain to user
- Accuracy & Scalability questionable



Reference: From the lecture slide "Recommender Systems" by Alex Smola

The limitation of K-NN recommendation.

- The similarity between users or items are under-estimated due to **data sparsity**.
  - User A and User B will be a neighbor only if they share significant amount of purchase history.
  - However, the histories of users are naturally very sparse, and thus users can have different histories even though they have similar preference

The sparsity problem

- Netflix dataset
  - The number of users: 500k
  - The number of items: 17k
  - The total number of possible ratings
    - 500k X 17k = 8.5 B
  - The total number of actual ratings = 10 M
  - The portion of non-zero entries = **0.11%**

Matrix-factorization based recommendation

Model-Based Recommendation

Big Data

Model-based recommendation techniques

- **Question** – How to avoid sparsity problem?
- **Solution** – use latent model
  - Compress the vector into a dense lower-dimensional space (latent space) where well preserves the similarity between users and items
  - Compute the similarity between users and items in the latent space

What is latent model?

- Simple quiz: Cluster the following animals by three groups

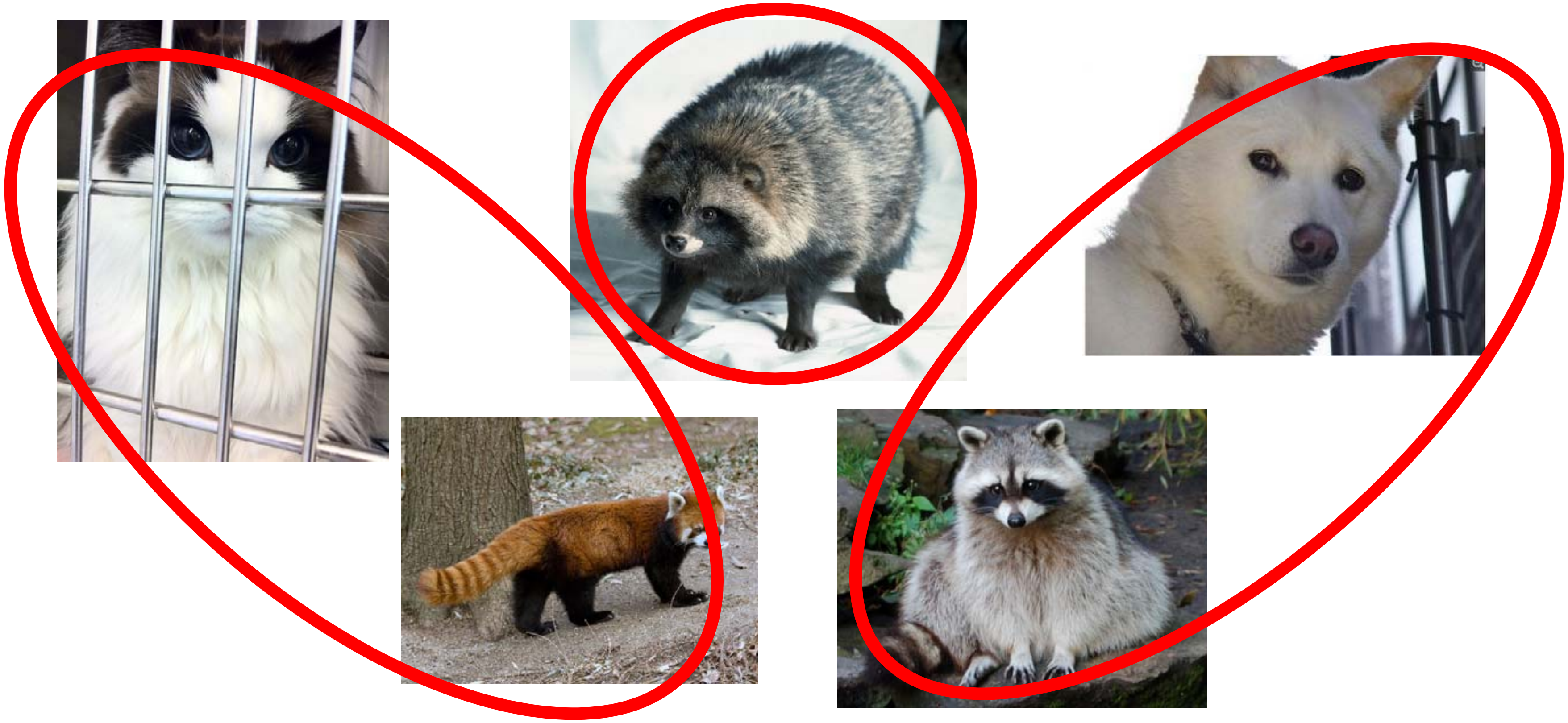




What is latent model?

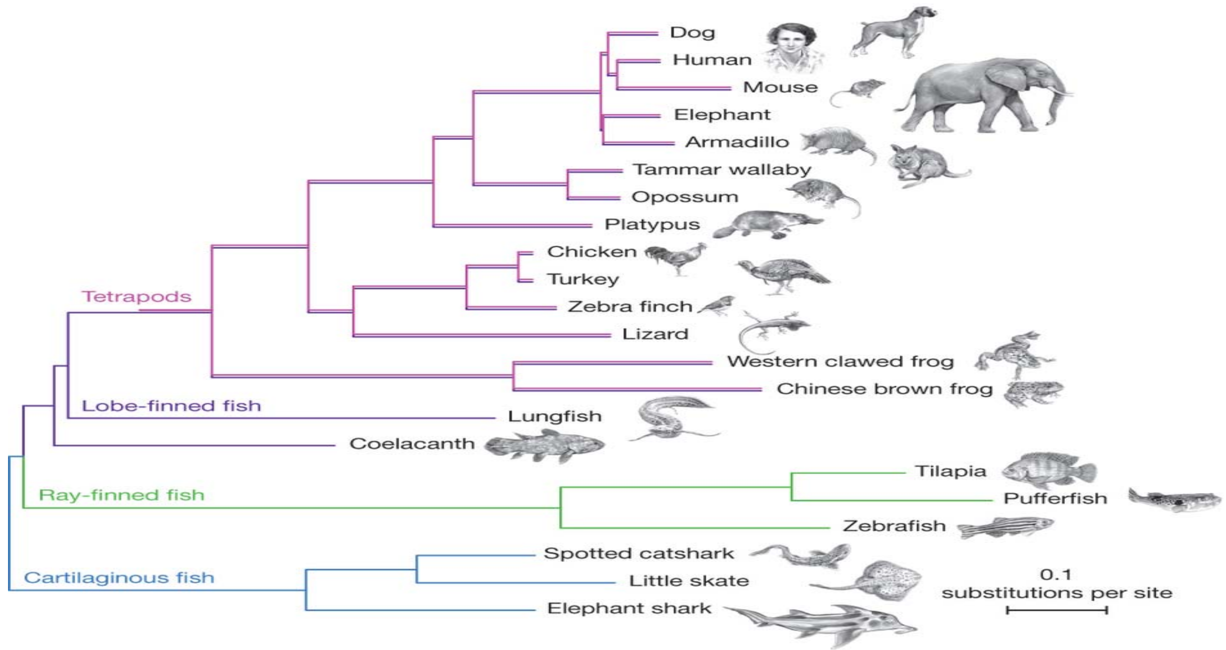
- Simple quiz: Cluster the following animals by three groups

However, the correct answer is…



Phylogenetic tree: A latent model for animal

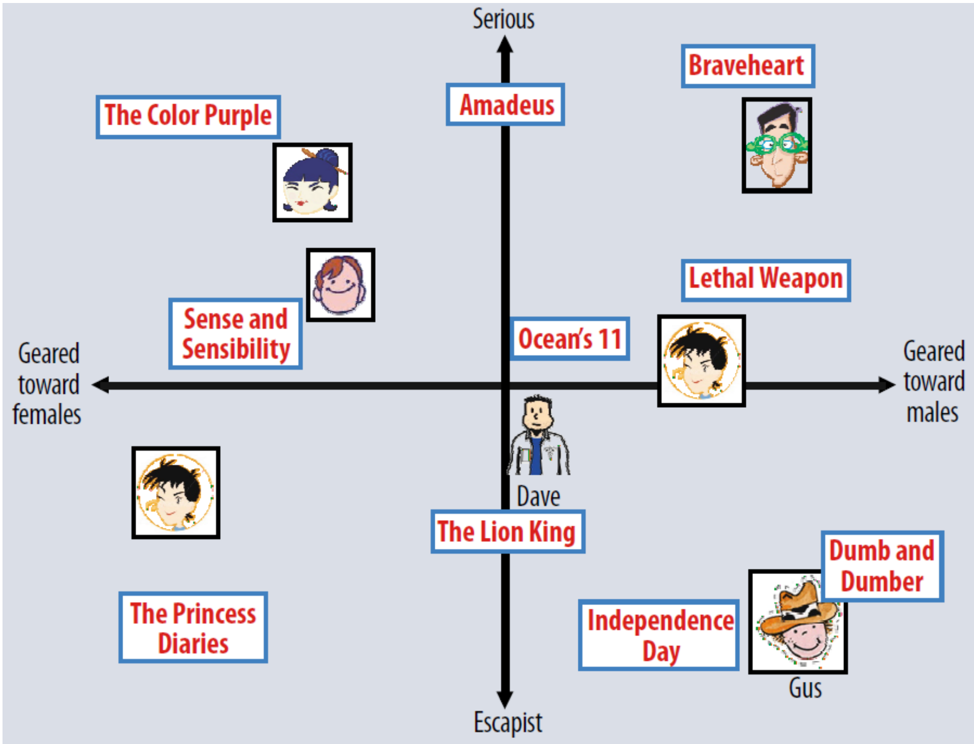
- This information is hidden, but it provides richer information for the animals, and enable us to get deep understand for their habits
- Latent model is a hidden model which well describes phenomena



Latent model for recommendation

- Users and Items can be represented as vectors in the shared latent space
- Rating score is generated by inner product of user latent vector (p_i) and item latent vector (q_j)
  - $\widehat{\mathrm{r}_{ij}} = \mathrm{p}_i^T \cdot \mathrm{q}_j$

Example of mapping users and items on a latent space

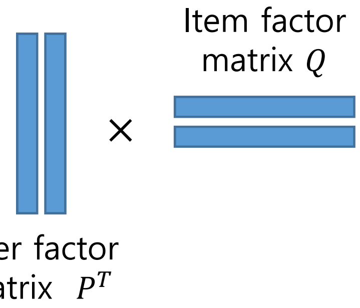


[Koren09]

Formal description

- Latent Model

$$\begin{array}{c} \\ Alice \\ Bob \\ Charlie \end{array} \begin{array}{c} \begin{array}{ccc} Avatar & The\ Matrix & Up \end{array} \\ \begin{pmatrix} ? & 4 & 2 \\ 3 & 2 & ? \\ 5 & ? & 3 \end{pmatrix} \end{array} \approx$$



Original matrix R

  - $r_{ij} \approx \widehat{r_{ij}} = [P^T Q]_{ij}$

- Goal: Find P and Q which minimize the error (RMSE)

  - $\underset{P,Q}{\operatorname{argmin}}\, RMSE(R, P^T Q)$