Introduction to Recommender Systems

Isabelle Augenstein
augenstein@di.ku.dk
@IAugenstein
http://isabelleaugenstein.github.io/

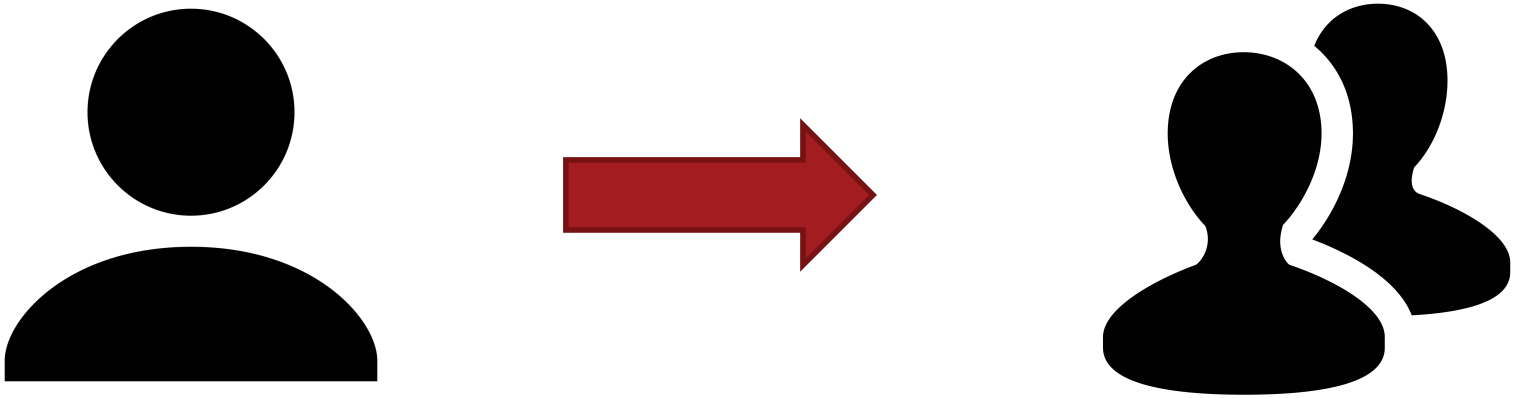
Web Science Course
26 February 2019

UNIVERSITY OF COPENHAGEN

This Lecture

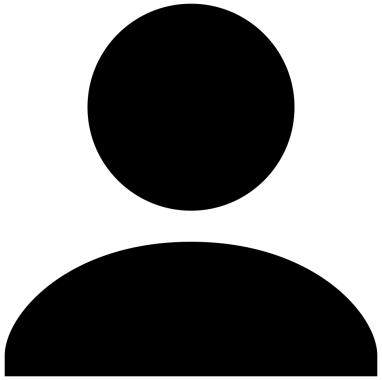
- Problem Setting
- Real-World Examples
- Basic Representations and Scoring
- Evaluation

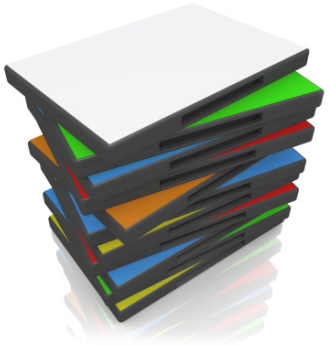
Example 1



tinder

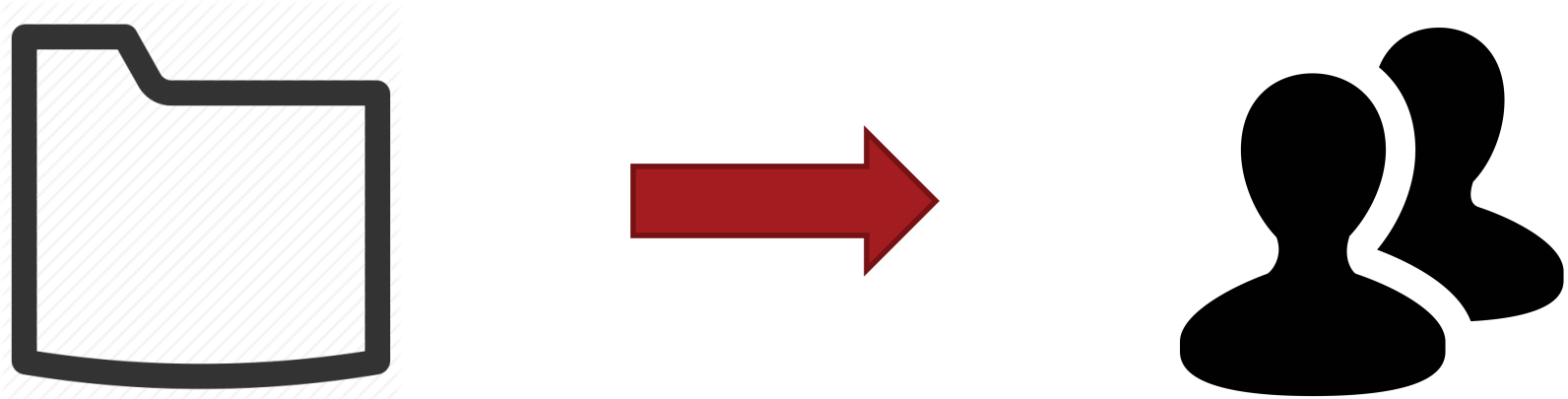
Example 2



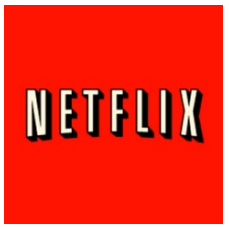


NETFLIX

Example 3



Many Real-World Examples



Problem Setting

	Fargo	Splash	Titanic	Frozen	Juno	Gladiator
John	4				5	1
Mary	5		4			
Lars				2	4	5
Mette		3		3		

Utility / Ratings matrix: user-item pairs & their ratings

$$Y : U \times M \rightarrow R$$

Non-Personalised Recommendation

- Popularity-Based Recommendation
- Demographics-Based Recommendation
- Item Co-Occurrence

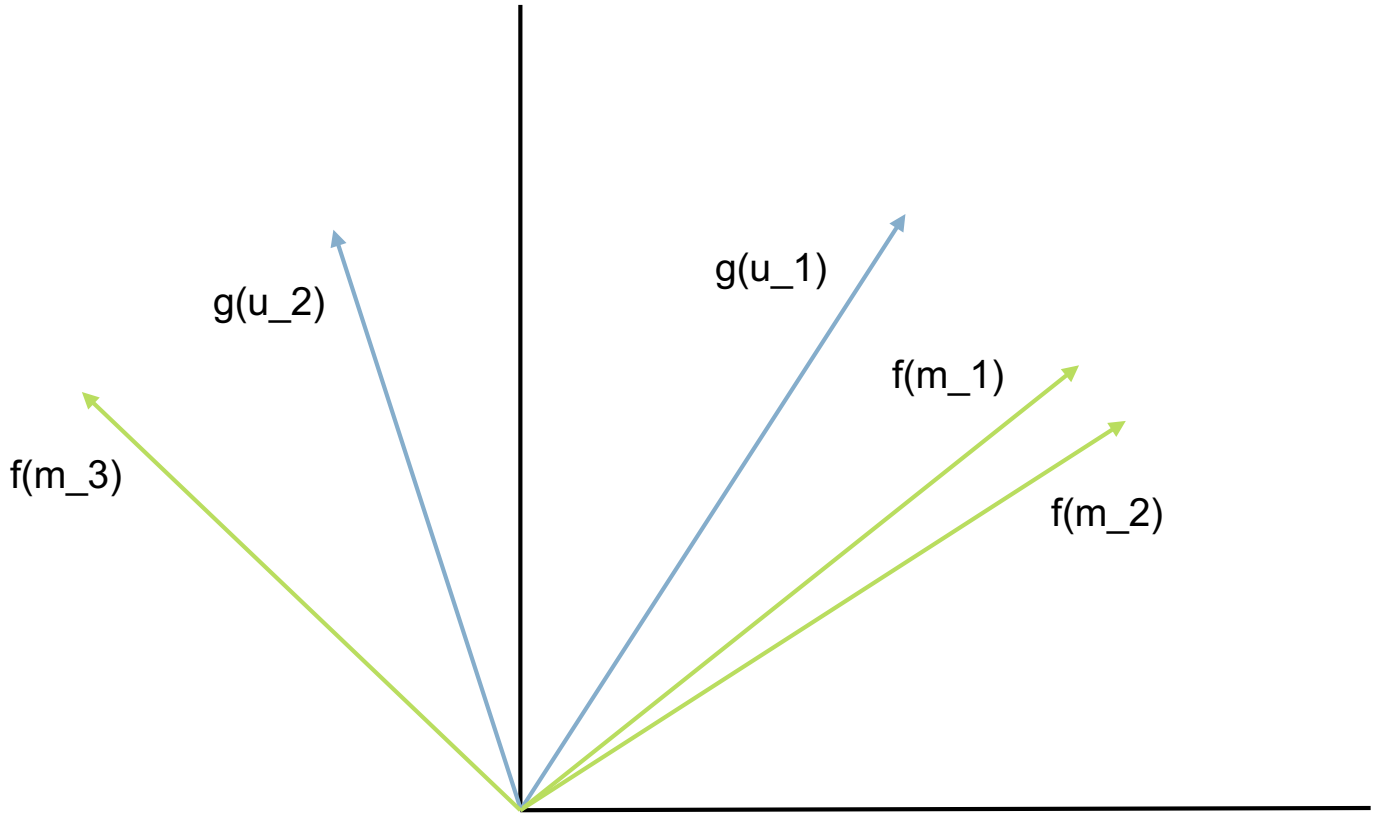
Problem Setting: Content-Based Filtering

Fargo: *Fargo is a 1996 black comedy-crime film written, produced, and directed by Joel and Ethan Coen. Frances McDormand stars as (…)*

Splash: *Splash is a 1984 American fantasy romantic comedy film directed by Ron Howard, written by Lowell Ganz and Babaloo Mandel, and starring (…)*

Titantic: *Titanic is a 1997 American epic romance and disaster film directed, written, co-produced and co-edited by James Cameron. A fictionalized (…)*

Content-Based Filtering: Overall Approach



Distance Function: d(f(m_i), g(m_j) = d(x_i, x_j)

Content-Based Filtering: Vector Construction

TF-IDF score of word *t* in document *d*:

$$tf_{t,d} \cdot idf_{t,D} = \frac{n_{t,d}}{\sum_{t' \in d} n_{t',d}} \cdot log \frac{|D|}{|\{d \in D : t \in d\}|}$$

$n_{t,d}$: number of times term *t* occurs in document *d*

$|D|$: number of documents in dataset

$|\{d \in D : t \in d\}|$: number of documents where *t* appears

→ Words of highest TF-IDF 'characterise' the document

Vector Distance Functions

Euclidian Distance

$$d([x_1, x_2, \ldots, x_n], [y_1, y_2, \ldots, y_n]) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Vector Distance Functions

Most Common: Cosine Similarity

$$\theta([x_1, x_2, \ldots, x_n], [y_1, y_2, \ldots, y_n]) = \frac{x \circ y}{\sqrt{x \circ x}\sqrt{y \circ y}}$$

- $\circ$ signifies dot product

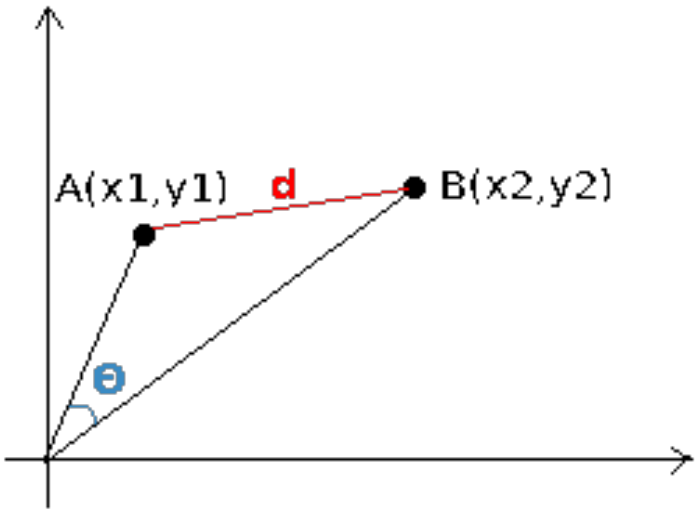
Vector Distance Functions

Euclidian Distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Cosine Similarity

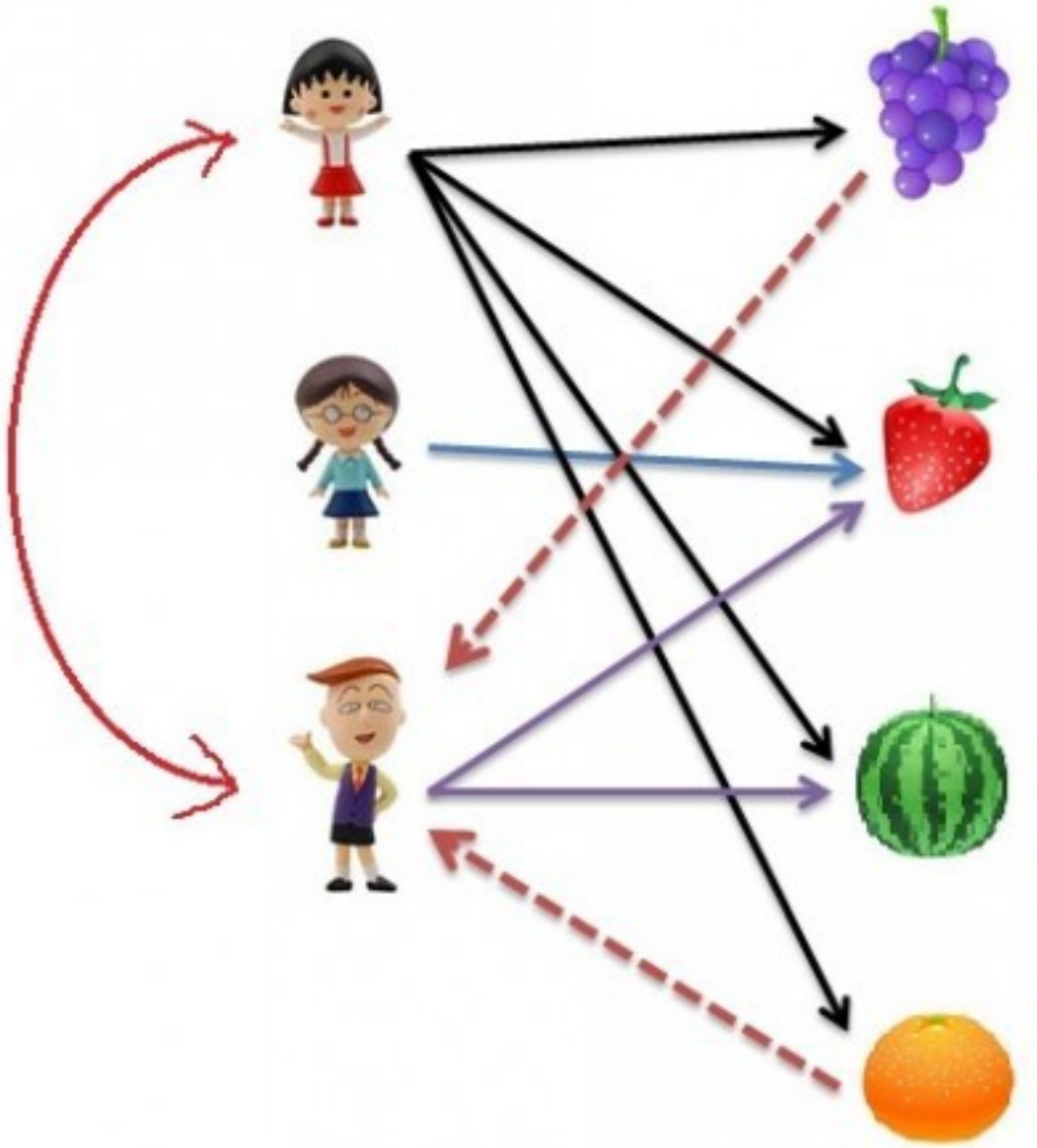
$$\theta(x, y) = \frac{x \circ y}{\sqrt{x \circ x}\sqrt{y \circ y}}$$



Euclidian vs. Cosine: https://cmry.github.io/notes/euclidean-v-cosine

User-Based Collaborative Filtering

- Step 1: Users are similar if they share similar items
- Step 2: Items from similar users are good recommendations



User-Based Collaborative Filtering Example

	i_1	i_2	i_3	i_4	i_5
u_1	8	1	?	2	7
u_2	2	?	5	7	5
u_3	5	4	7	4	7
u_4	7	1	7	3	8
u_5	1	7	4	6	5
u_6	8	3	8	3	7

- Count-Based

$$f(u_l, u_k) = \frac{\sum_j |r(u_l, i_j) - r(u_k, i_j)|}{\#\ items}$$

- Pearson Correlation

$$f(u_l, u_k) = \frac{cov(u_l, u_k)}{\sigma(u_l)\sigma(u_k)}$$

- How similar are u_1 and u_2?
- How similar are u_1 and u_5?

Collaborative Filtering Scoring Functions

- Count-Based
- Pearson Correlation
- L2-Normalised Euclidian Distance
- Cosine Similarity

Collaborative Filtering Scoring Functions

- Count-Based

$$f(u_l, u_k) = \frac{\sum_j |r(u_l, i_j) - r(u_k, i_j)|}{\#\ items}$$

- Pearson Correlation

$$f(u_l, u_k) = \frac{cov(u_l, u_k)}{\sigma(u_l)\sigma(u_k)}$$

- Euclidian Distance

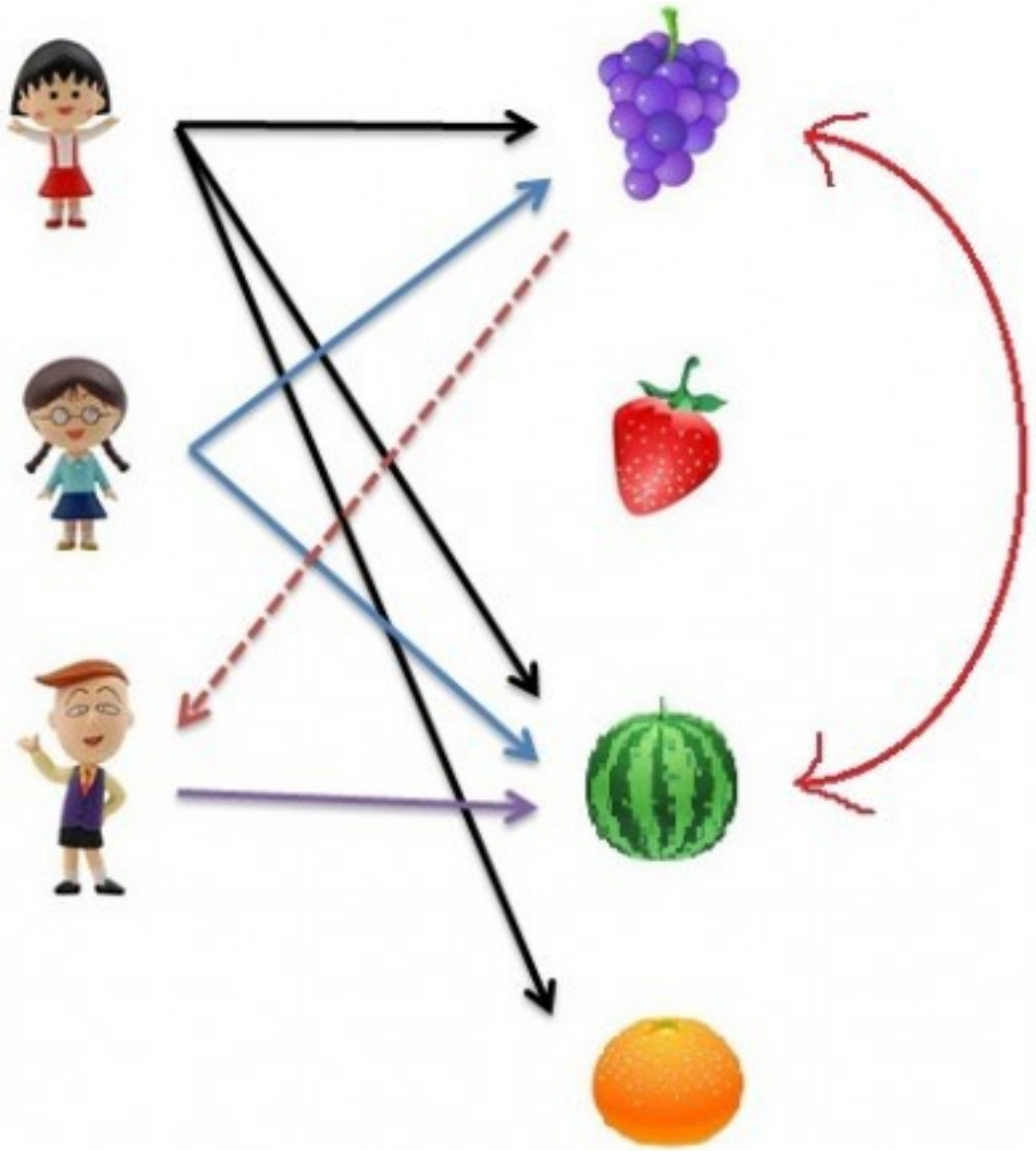
$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Cosine Similarity

$$\theta(x, y) = \frac{x \circ y}{\sqrt{x \circ x}\sqrt{y \circ y}}$$

Item-Based Collaborative Filtering

- Step 1: Items are similar if they share similar users
- Step 2: Items more similar to the user's items are good recommendations



Item-Based Collaborative Filtering Example

	i_1	i_2	i_3
u_1	2	?	3
u_2	5	2	?
u_3	3	3	1
u_4	?	2	2

- How similar are u_1 and u_2?
- How similar are u_1 and u_3?
- How similar are u_2 and u_3?

User vs. Item-Based Collaborative Filtering

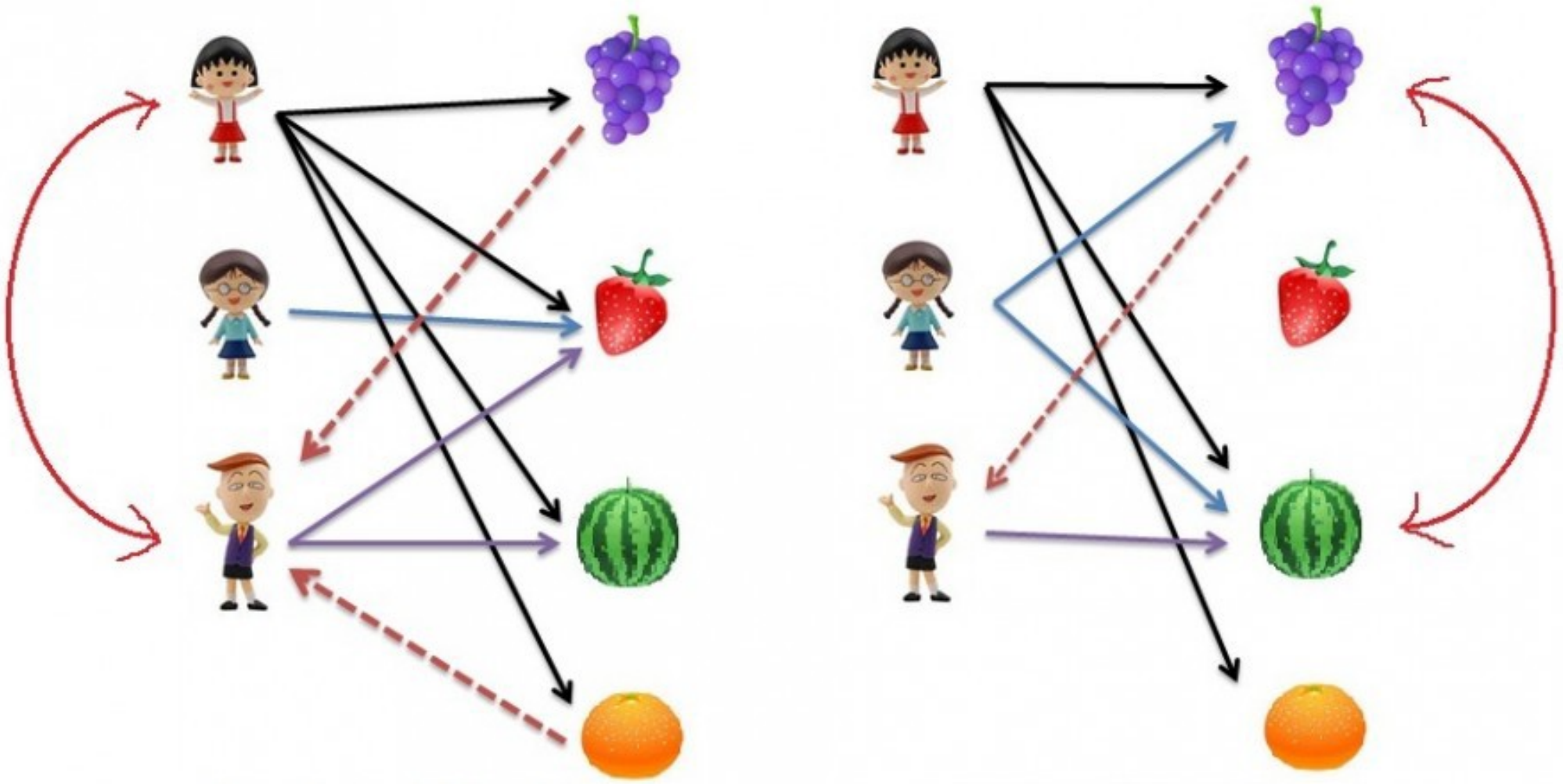


Image credit: https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f

Personalised Recommendation

- Content-Based Filtering
- Collaborative Filtering
  - User-Based
  - Item-Based

Evaluation

- Mean Absolute Error
- Root Mean Squared Error
- Precision, Recall, F1

Evaluation: Mean Absolute Error

$$MAE = \frac{\sum_{(i,j)} \left| p(u_i, m_j) - r(u_i, m_j) \right|}{n}$$

n: number of ratings of items over all users
$p(u_i, m_j)$: predicted rating of user u_i on item m_j
$r(u_i, m_j)$: gold rating

lower = better

Evaluation: Root Mean Squared Error

$$RMSE = \sqrt{\frac{1}{n}\sum_{(i,j)}^{N}(p(u_i, m_j) - r(u_i, m_j))^2}$$

n: number of ratings of items over all users
$p(u_i, m_j)$: predicted rating of user u_i on item m_j
$r(u_i, m_j)$: gold rating

lower = better

Evaluation: Precision, Recall, F1

	Relevant	**Irrelevant**
Recommended	TP	FP
Not recommended	FN	TN

Precision (P): $\frac{TP}{TP+FP}$; *Recall (R)*: $\frac{TP}{TP+FN}$

$$F1 = \frac{2PR}{P+R}$$

higher = better

Evaluation

- Mean Absolute Error
- Root Mean Squared Error
- Precision, Recall, F1

Summary – This Lecture

- Problem Setting
  - Real-World Examples
- Basic Representations and Scoring
  - Non-Personalised Recommendation
    - Popularity-Based Recommendation
    - Demographics-Based Recommendation
    - Item Co-Occurrence
  - Personalised Recommendation
    - Content-Based Filtering
    - Collaborative Filtering
      - User-Based
      - Item-Based
- Evaluation

Next Lecture

- Machine Learning Based Recommender Systems
- Latent User and Item Representations
- Brief Introduction to Representation Learning
- Context-Dependent Recommendation

Thank you!

augenstein@di.ku.dk
@IAugenstein
http://isabelleaugenstein.github.io/