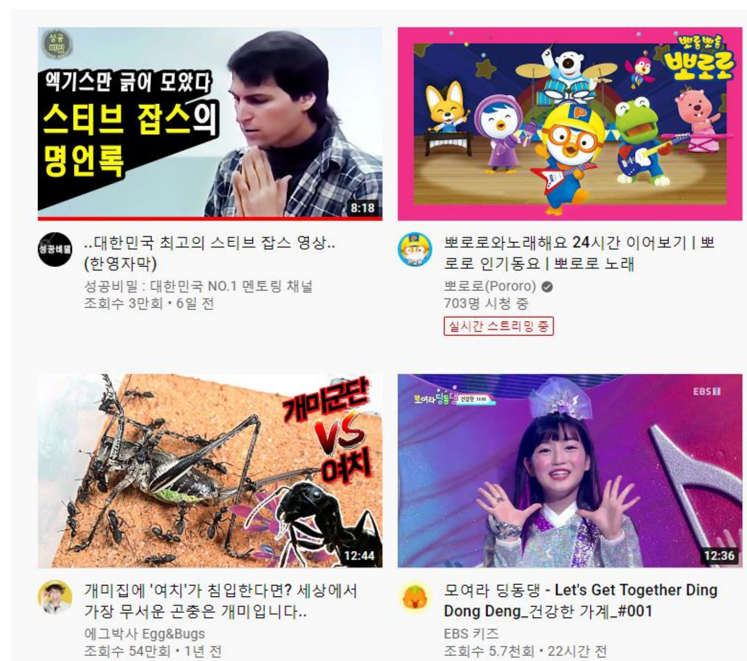
Service Intelligence Week 2.

[Recommender Systems for Services]

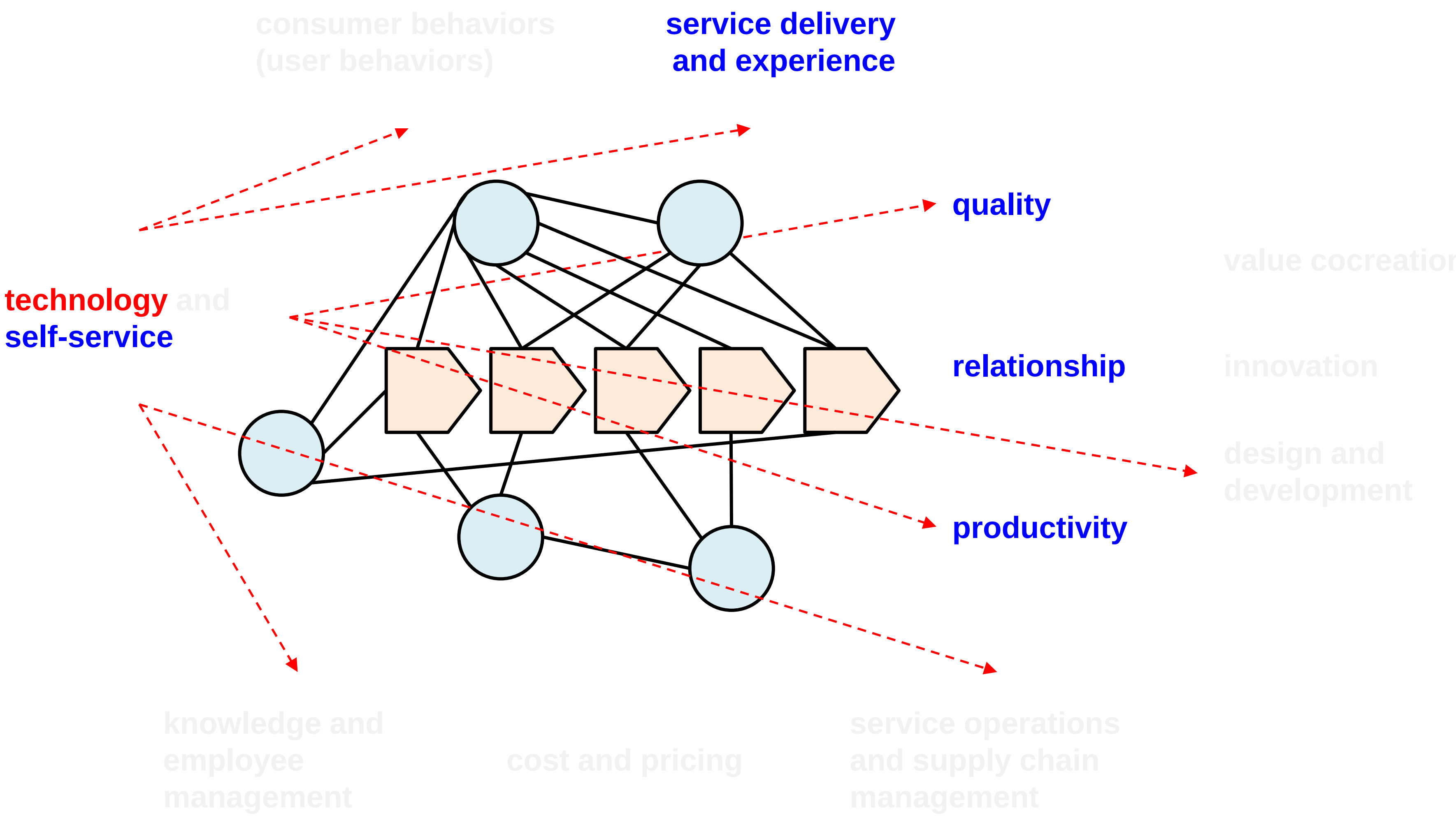
Chiehyeon Lim

2022. 9. 5

Recommender Systems for Services

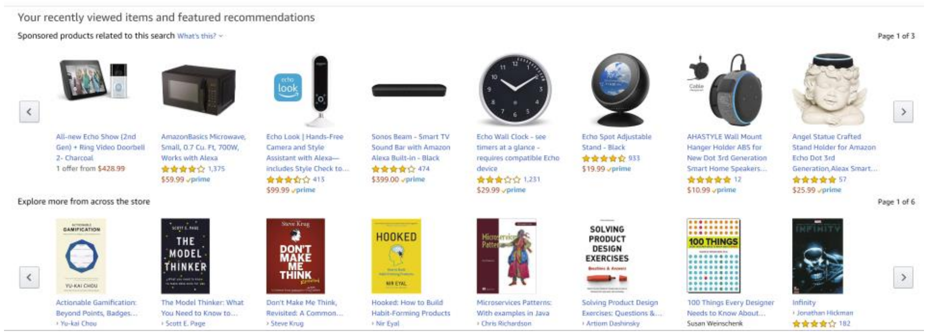


Recommender Systems for Services



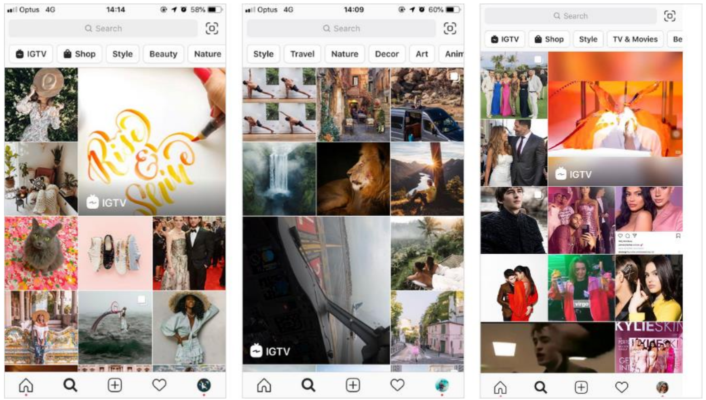
Recommender Systems for Services

- Amazon



  - Recommend items based on the user's purchase records, item characteristics, and search queries
  - Expose the recommended items in the browser continuously to enhance the click through rate

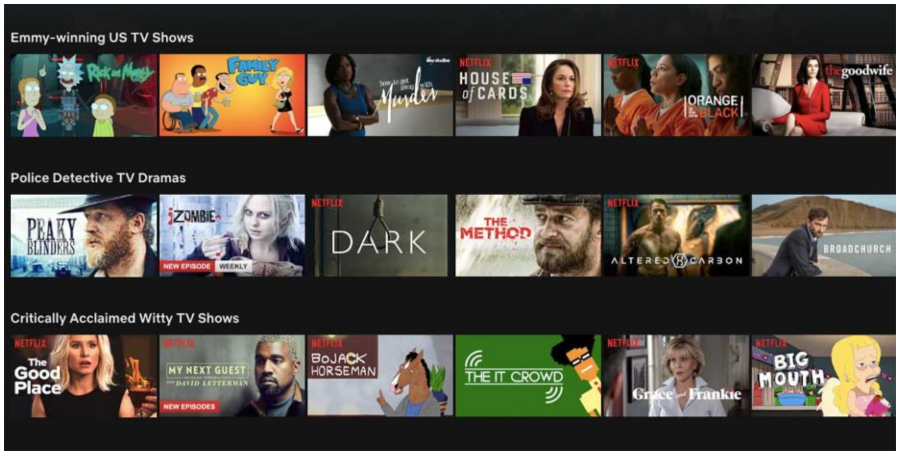
- Instagram



  - Recommend based on the users' historical activities and interactions with other users
  - Focus on increasing the service use time

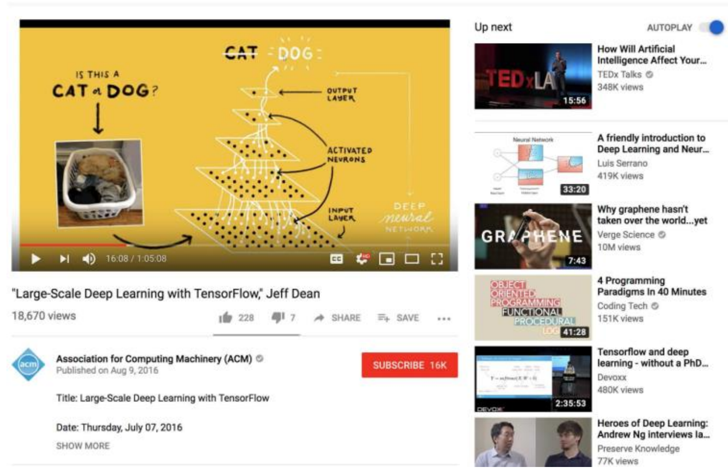
Recommender Systems for Services

■ Netflix



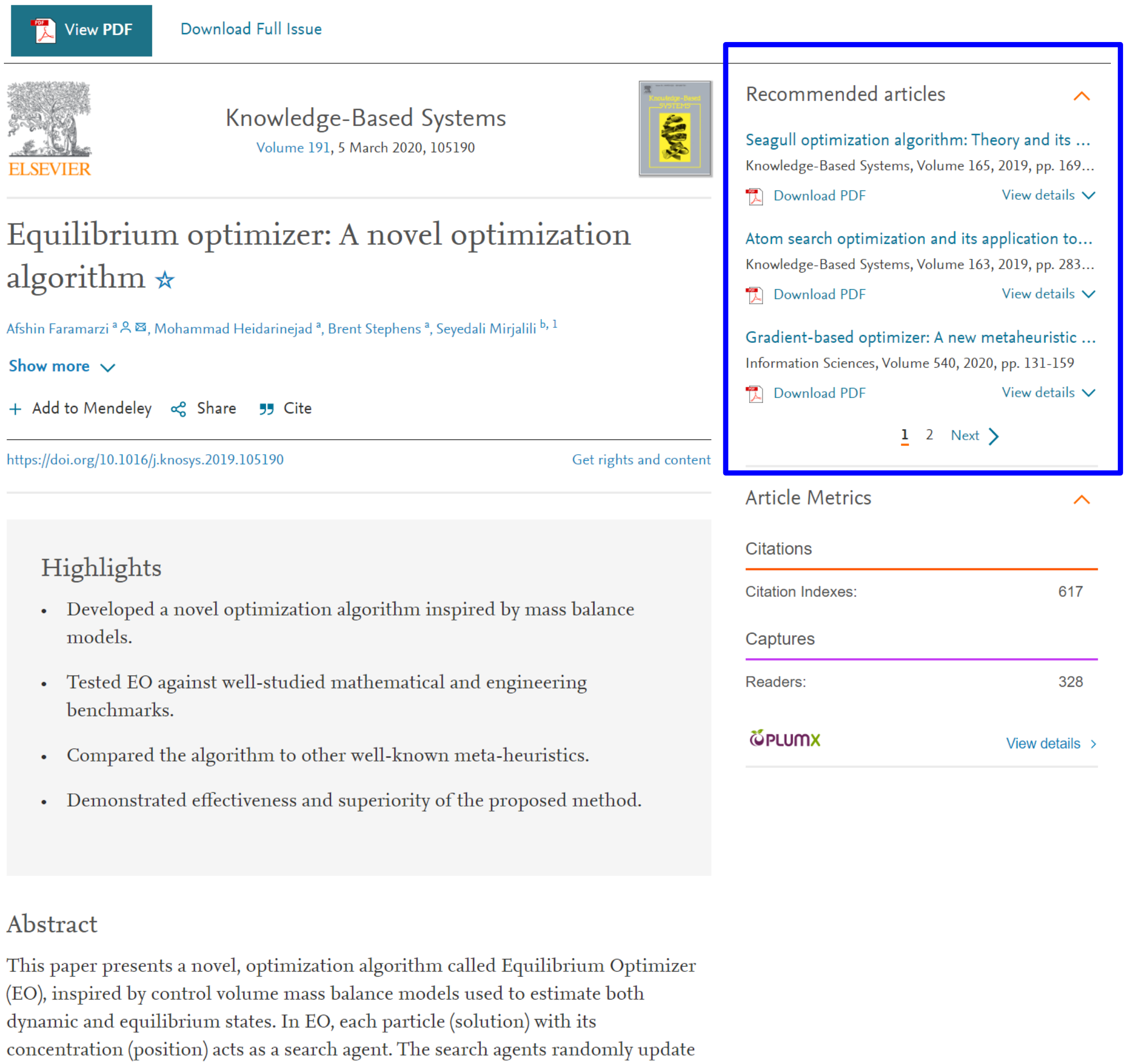
- Combines multiple recommender systems to fit the diverse contexts and enhance the success rate of recommendation
- Focus on increasing the service use time

■ Youtube



- Combines different recommender systems for the initial recommendation in the first page and for the recommendation while watching the video
- Consider the exposure of diverse contents

Recommender Systems for Services



Why Recommender Systems for Services?

- Contribution and impact of recommender systems to the click through rate and actual purchase

Amazon	YouTube	Netflix	Instagram
35%	**70%**	**75%**	**50%**
Proportion of the purchase based on recommendation	Proportion of the watch based on recommendation	Proportion of the watch based on recommendation	Proportion of the users who want recommendation

- Recys contribute to the positive service experience, sales, and customer loyalty (Gomez-Uribe et al., 2015)
- Recys effectively reflect the value from items that users look for (Schafer et al., 1999)
- Recys help users make a better decision that fits to their contexts and needs (Vig et al., 2009)

Why Recommender Systems for Services?

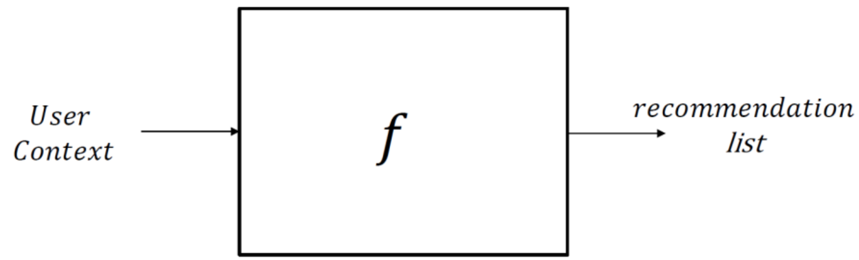
- Service is simply to serve customers: Help tasks or to do the tasks

  Service becomes effective, when there is a capability/effort gap between the provider and customer

- "Humans are facing an increasing number of choices in every aspect of their lives—certainly around media such as videos, music, and books, other taste-based questions such as vacation rentals, restaurants, and so on, but more importantly, around areas such as health insurance plans and treatments and tests, job searches, education and learning, dating and finding life partners, and many other areas in which choice matters significantly. We are convinced that the field of recommender systems will continue to play a pivotal role in using the wealth of data now available to make these choices manageable, effectively guiding people to the truly best few options for them to be evaluated, resulting in better decisions" (Gomez-Uribe and Hunt, 2015; The Netflix Recommender System - Algorithms, Business Value, and Innovation)

Recommender Systems for Services

$$y = f(x)$$



recommend

verb [T]

UK /ˌrek.əˈmend/ US /ˌrek.əˈmend/

B1

to suggest that someone or something would be good or suitable for a particular job or purpose, or to suggest that a particular action should be done:

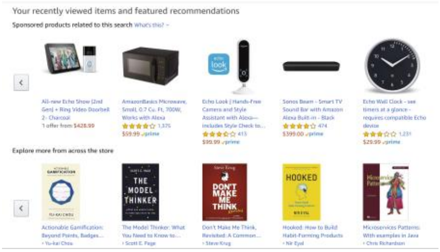
- *I can recommend the chicken in mushroom sauce - it's delicious.*
- *She has been recommended* ***for*** *promotion.*
- *The headmistress agreed to recommend the teachers' proposals* ***to*** *the school governors.*
- [+ (that)] *The doctor recommended* ***(that)*** *I get more exercise.*
- [+ -ing verb] *I recommend writ****ing*** *your feelings down on paper.*
- *The city* ***has much/little to*** *recommend it (= it has many/few pleasant qualities).*

Reference: https://dictionary.cambridge.org/dictionary/english/recommend

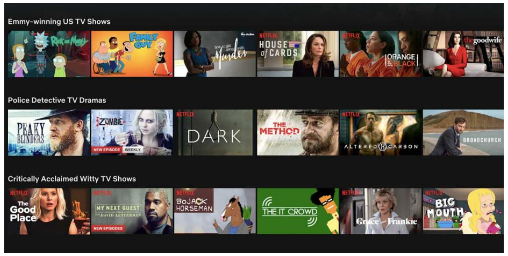
Approaches of Recommender Systems: A Categorization

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data
  - Content-based filtering
    - Analyzes a set of documents (of the items in question) rated by an individual user and uses the contents of the documents, as well as the provided ratings, to infer a user profile that can be used to recommend additional items of interest
    - There is an overspecialized recommendations problem
  - Collaborative filtering
    - Uses an information filtering technique based on the users' previous evaluation of items or history of previous purchases
    - There is a sparsity problem

amazon



NETFLIX

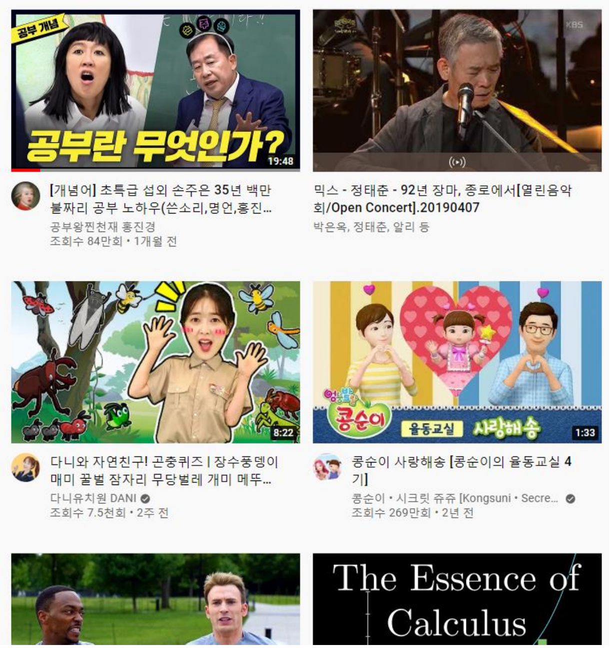


Approaches of Recommender Systems: A Categorization

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data
  - Content-based filtering
  - Collaborative filtering

Approaches of Recommender Systems: A Categorization

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data
  - Content-based filtering
  - Collaborative filtering

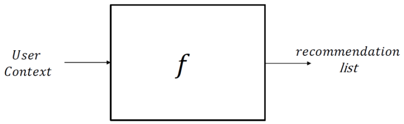


Approaches of Recommender Systems: A Data Perspective

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data

	Feature 1	Feature 2	Feature 3	…	Feature m-1	Feature m
Transaction 1	…	…	…	…	…	…
Transaction 2	…	…	…	…	…	…
Transaction 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
Transaction n-1	…	…	…	…	…	…
Transaction n	…	…	…	…	…	…

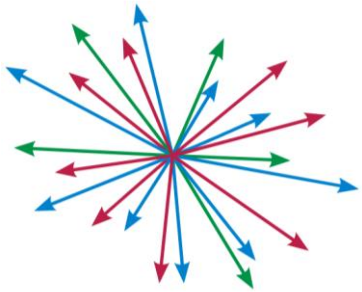
$$y = f(x)$$

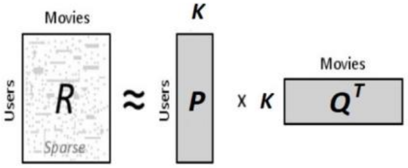


Approaches of Recommender Systems: A Data Perspective

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data

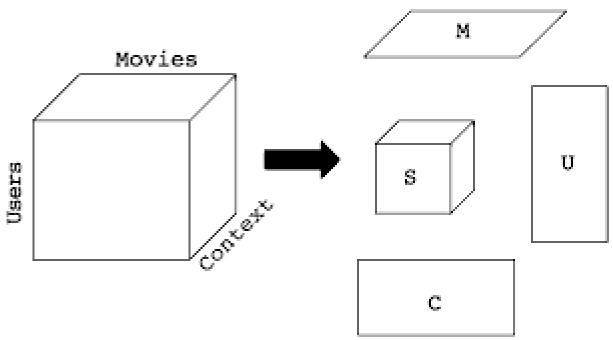
	Item 1	Item 2	Item 3	…	Item m-1	Item m
User 1	…	…	…	…	…	…
User 2	…	…	…	…	…	…
User 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
User n-1	…	…	…	…	…	…
User n	…	…	…	…	…	…

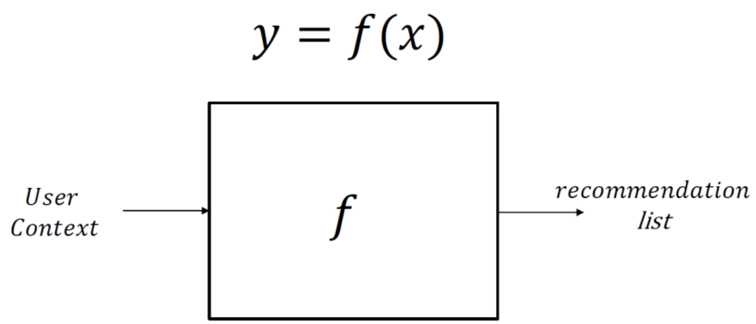




Approaches of Recommender Systems: A Data Perspective

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data



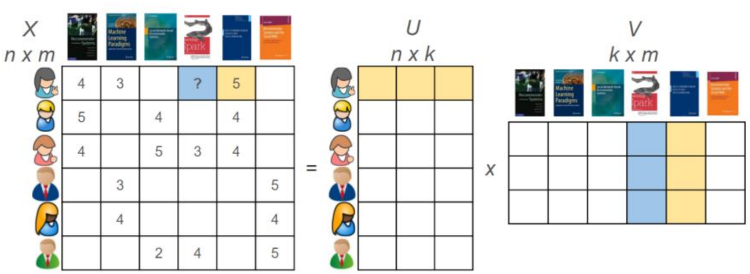


Approaches of Recommender Systems: Our Focus

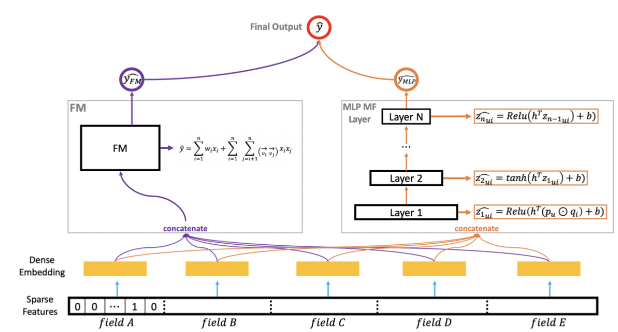
- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data

	Item 1	Item 2	Item 3	…	Item m-1	Item m
User 1	…	…	…	…	…	…
User 2	…	…	…	…	…	…
User 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
User n-1	…	…	…	…	…	…
User n	…	…	…	…	…	…

	Feature 1	Feature 2	Feature 3	…	Feature m-1	Feature m
Transaction 1	…	…	…	…	…	…
Transaction 2	…	…	…	…	…	…
Transaction 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
Transaction n-1	…	…	…	…	…	…
Transaction n	…	…	…	…	…	…



$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j$$



Traditional Approaches and the Use of the User Item Matrix

Traditional Approaches: Association Rule Mining

$$Rule: X \Rightarrow Y$$

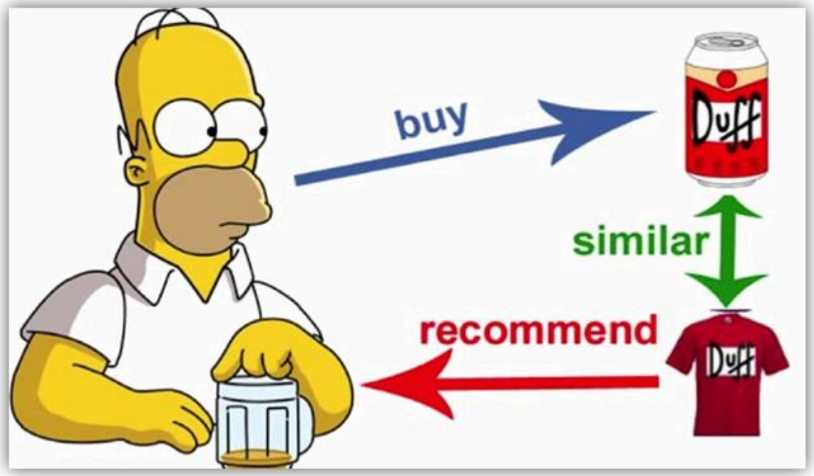
$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Image source: https://www.techleer.com/articles/238-concepts-behind-association-rule-mining/

Traditional Approaches: Contents-based Filtering



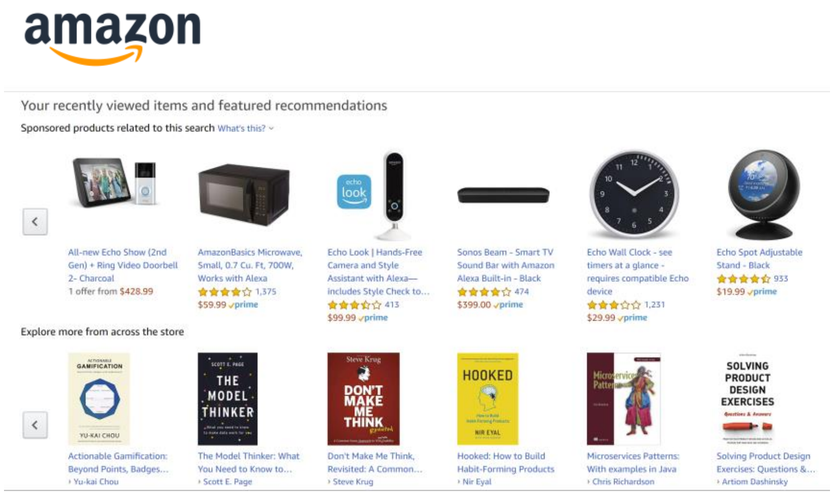
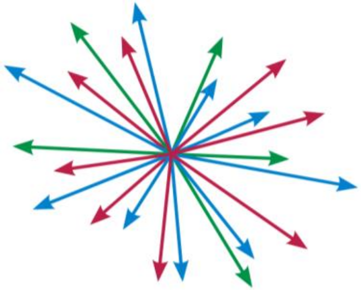


Image source: https://heartbeat.comet.ml/recommender-systems-with-python-part-i-content-based-filtering-5df4940bd831

Approaches of Recommender Systems: Collaborative Filtering

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data



	Item 1	Item 2	Item 3	…	Item m-1	Item m
User 1	…	…	…	…	…	…
User 2	…	…	…	…	…	…
User 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
User n-1	…	…	…	…	…	…
User n	…	…	…	…	…	…

Approaches of Recommender Systems: Collaborative Filtering

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data

	Item 1	Item 2	Item 3	…	Item m-1	Item m
User 1	…	…	…	…	…	…
User 2	…	…	…	…	…	…
User 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
User n-1	…	…	…	…	…	…
User n	…	…	…	…	…	…

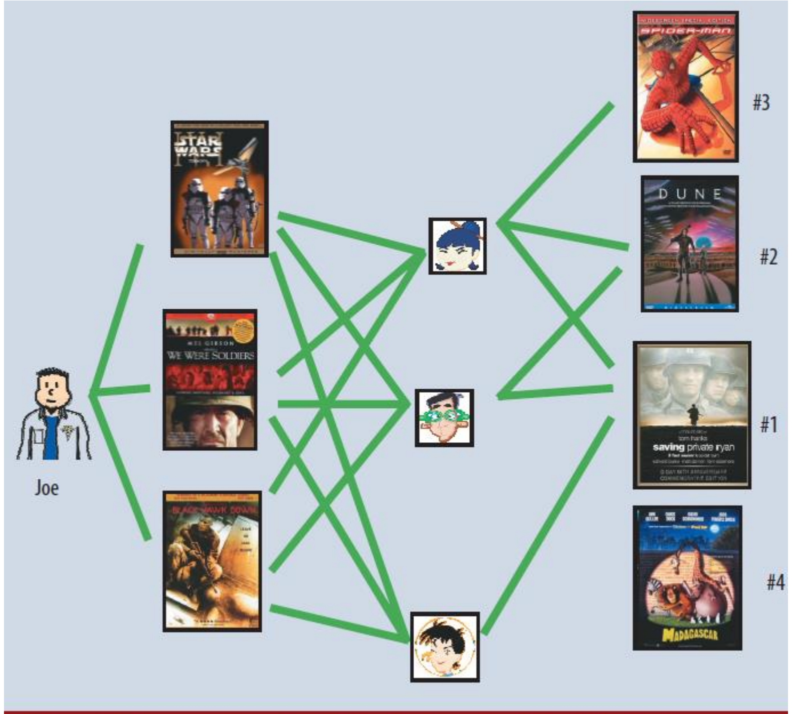
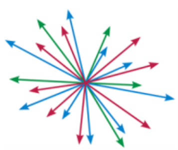


Figure 1. The user-oriented neighborhood method. Joe likes the three movies on the left. To make a prediction for him, the system finds similar users who also liked those movies, and then determines which other movies they liked. In this case, all three liked *Saving Private Ryan*, so that is the first recommendation. Two of them liked *Dune*, so that is next, and so on.

Image source: https://ieeexplore.ieee.org/document/5197422

Approaches of Collaborative Filtering: Similarity Measurement



	Item 1	Item 2	Item 3	…	Item m-1	Item m
User 1	…	…	…	…	…	…
User 2	…	…	…	…	…	…
User 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
User n-1	…	…	…	…	…	…
User n	…	…	…	…	…	…

	Item 1	Item 2	Item 3	…	Item m-1	Item m
User 1	…	…	…	…	…	…
User 2	…	…	…	…	…	…
User 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
User n-1	…	…	…	…	…	…
User n	…	…	…	…	…	…

- Euclidean distance
- Cosine similarity
- Jaccard coefficient
- Pearson correlation coefficient
- …

$$\mathrm{Sim}(u_a, u_b) = \frac{\sum_{i=1}^{i'} \left(r_{u_{a,i}}\right)\left(r_{u_{b,i}}\right)}{\sqrt{\sum_{i=1}^{n} \left(r_{u_{a,i}}\right)} \sqrt{\sum_{i=1}^{n} \left(r_{u_{b,i}}\right)}}$$

$$\mathrm{Sim}(u_a, u_b) = \frac{|I_{u_a}| \cap |I_{u_b}|}{|I_{u_a}| \cup |I_{u_b}|}$$

$$\mathrm{Sim}(u_a, u_b) = \frac{\sum_{i=1}^{i'} \left(r_{u_{a,i}} - r_{\overline{u_a}}\right)\left(r_{u_{b,i}} - r_{\overline{u_b}}\right)}{\sqrt{\sum_{i=1}^{i'} \left(r_{u_{a,i}} - r_{\overline{u_a}}\right)^2} \cdot \sqrt{\sum_{i=1}^{i'} \left(r_{\overline{u_{b,i}}} - r_{\overline{u_b}}\right)^2}}$$

Formula reference:
https://www.researchgate.net/publication/339466606_A_Survey_of_Similarity_Measures_for_Collaborative_Filtering-Based_Recommender_System

Approaches of Collaborative Filtering: Matrix Factorization

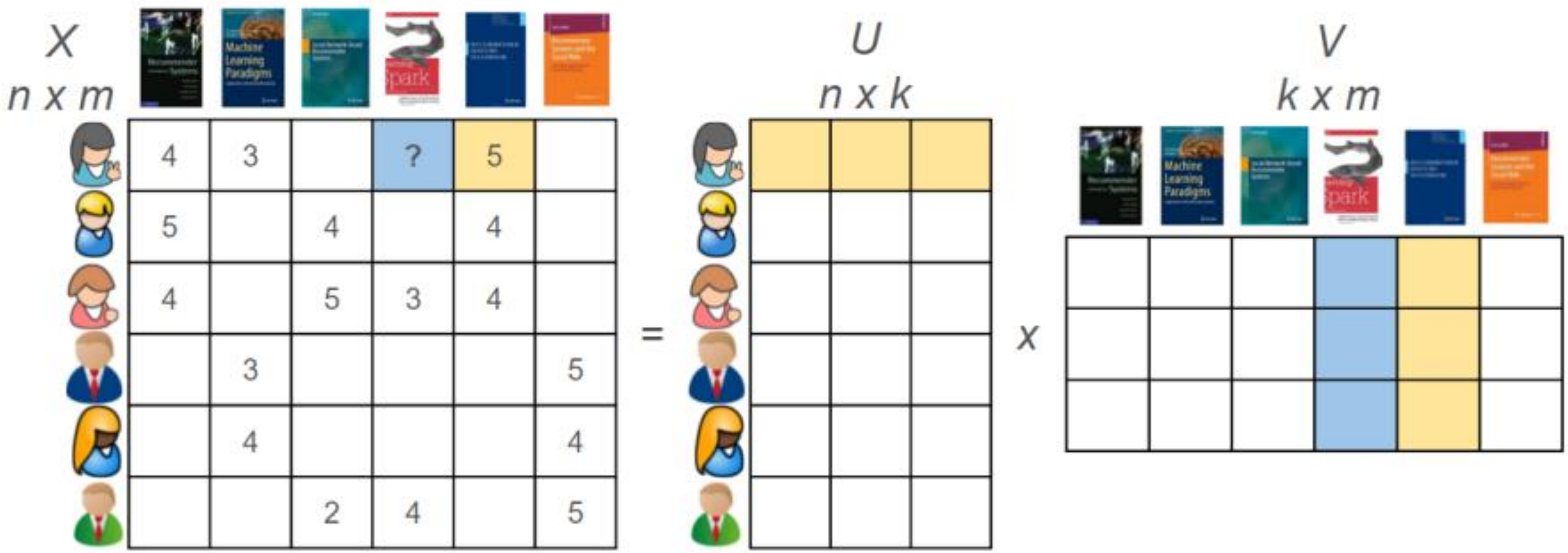
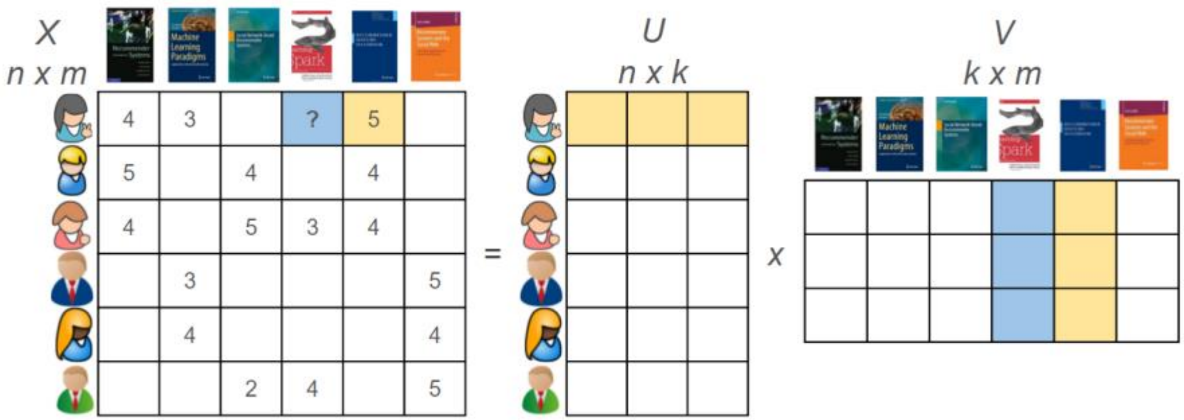
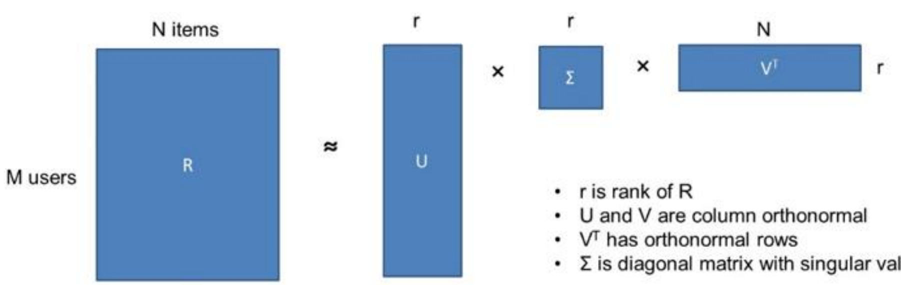


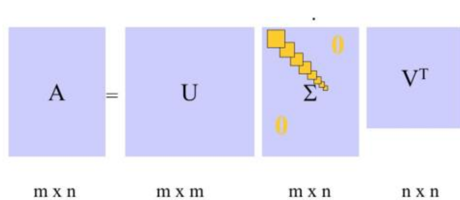
Image source: https://ieeexplore.ieee.org/document/5197422

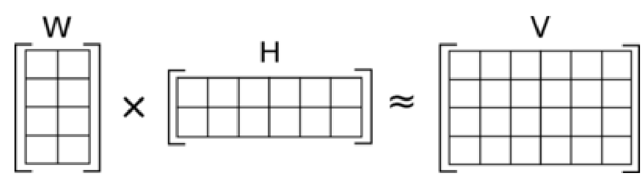
Approaches of Collaborative Filtering: Matrix Factorization





- r is rank of R
- U and V are column orthonormal
- V^T has orthonormal rows
- Σ is diagonal matrix with singular values





initialize: $\mathbf{W}$ and $\mathbf{H}$ non negative.

Then update the values in $\mathbf{W}$ and $\mathbf{H}$ by computing the following, with n as an index of the iteration.

$$\mathbf{H}^{n+1}_{[i,j]} \leftarrow \mathbf{H}^{n}_{[i,j]} \frac{((\mathbf{W}^n)^T\mathbf{V})_{[i,j]}}{((\mathbf{W}^n)^T\mathbf{W}^n\mathbf{H}^n)_{[i,j]}}$$

and

$$\mathbf{W}^{n+1}_{[i,j]} \leftarrow \mathbf{W}^{n}_{[i,j]} \frac{(\mathbf{V}(\mathbf{H}^{n+1})^T)_{[i,j]}}{(\mathbf{W}^n\mathbf{H}^{n+1}(\mathbf{H}^{n+1})^T)_{[i,j]}}$$

Until $\mathbf{W}$ and $\mathbf{H}$ are stable.

Target rating matrix *R*

user \ item	1	2	3	4	5	6
1	5	?	4	?	3	3
2	?	4	?	?	3	?
3	3	?	?	3	?	?
4	?	4	?	5	?	1

User feature matrix *P* (initial state)

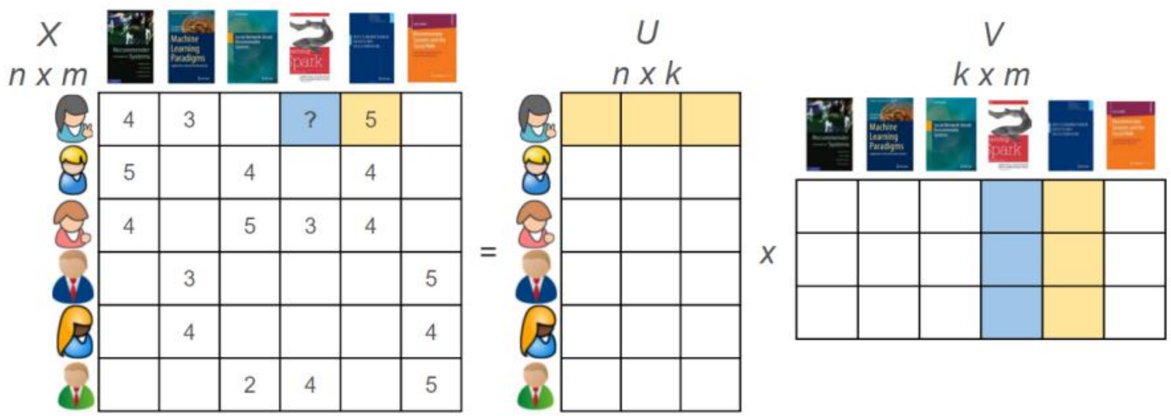
user \ dimension	1	2	3
1	0.11	0.07	0.19
2	0.09	0.16	0.19
3	0.09	0.05	0.04
4	0.03	0.13	0.18

Item feature matrix *Q* (initial state)

dimension \ item	1	2	3	4	5	6
1	0.16	0.01	0.07	0.17	0.02	0.20
2	0.18	0.19	0.10	0.05	0.18	0.15
3	0.02	0.18	0.03	0.14	0.17	0.06

Image source: https://medium.com/@lineyin/the-singular-value-decomposition-and-pca-cae825ff28fc
https://www.dataminingapps.com/2020/02/singular-value-decomposition-in-recommender-systems/
https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

Approaches of Collaborative Filtering: Matrix Factorization



$$\hat{r}_{ui} = q_i^T p_u. \qquad e_{ui} \stackrel{def}{=} r_{ui} - q_i^T p_u.$$

$$\min_{q^*, p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$$
$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u)$$

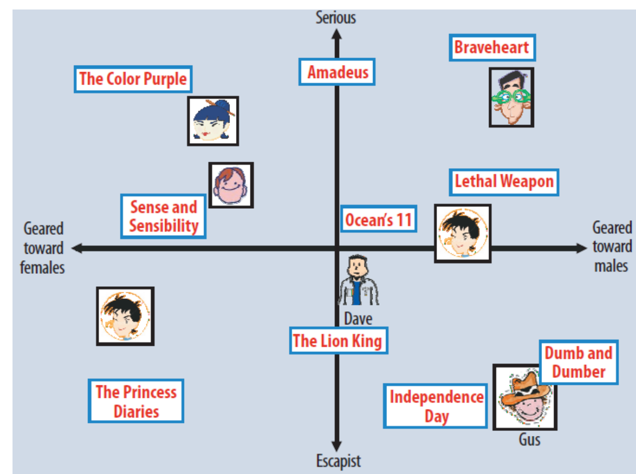
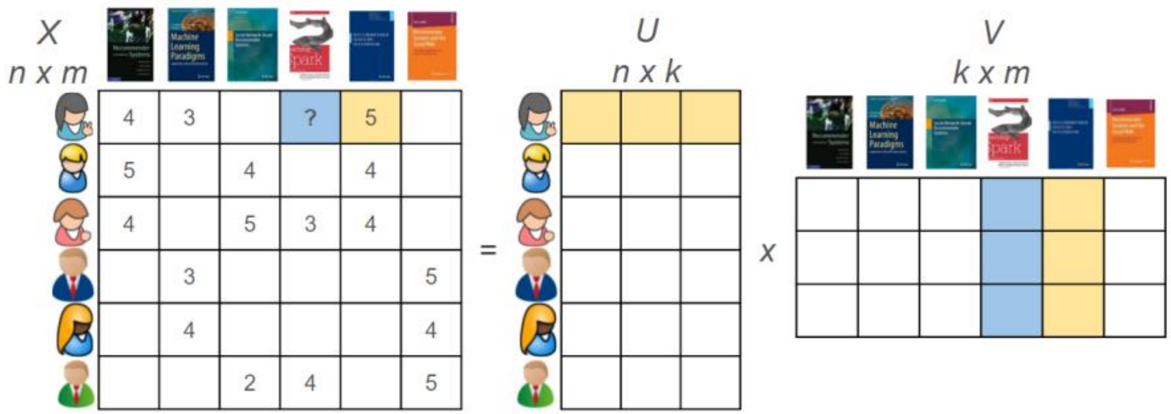


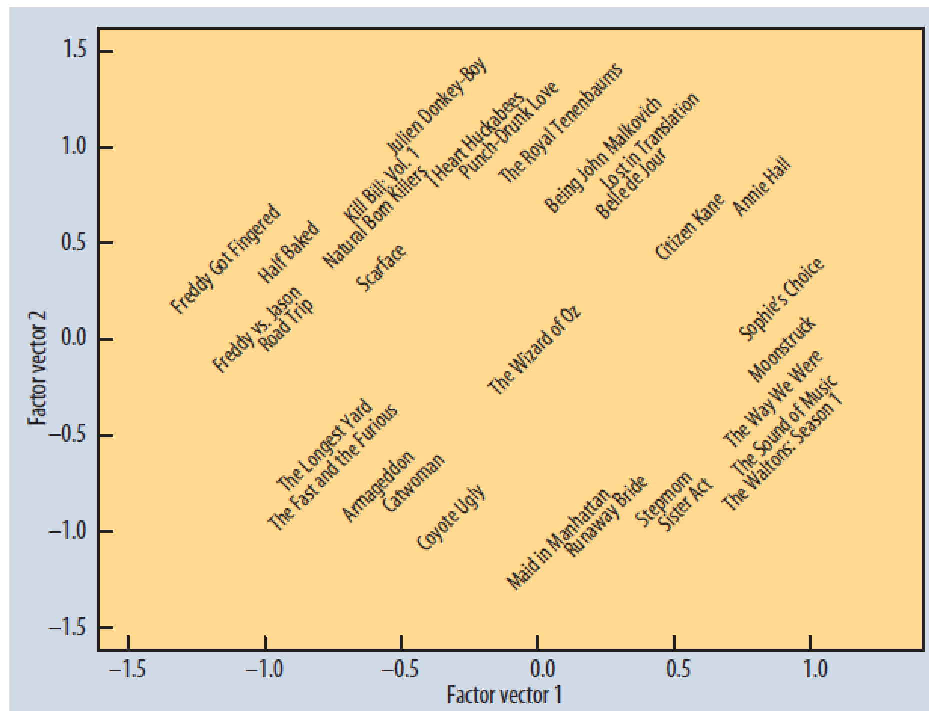
Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

Image source: https://ieeexplore.ieee.org/document/5197422

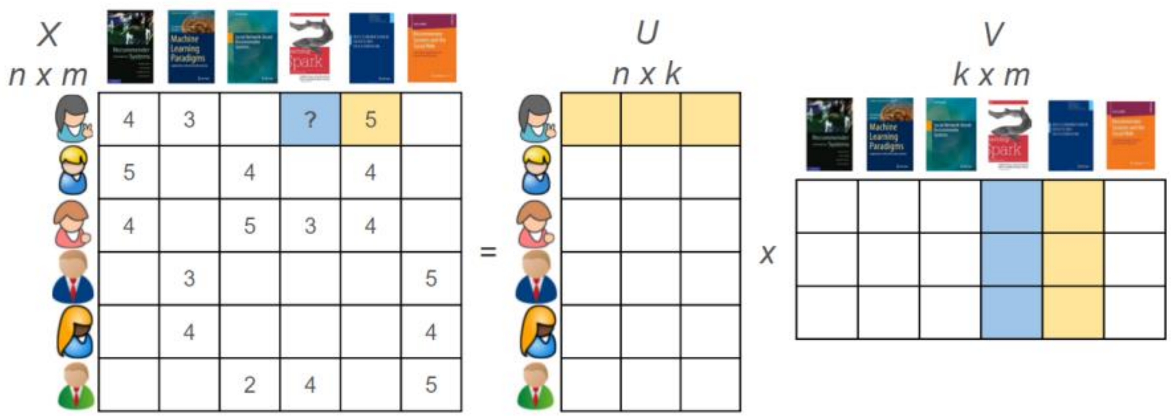
Approaches of Collaborative Filtering: Matrix Factorization



Factorizing the Netflix user-movie matrix allows us to discover the most descriptive dimensions for predicting movie preferences. We can identify the first few most important dimensions from a matrix decomposition and explore the movies' location in this new space. Figure 3 shows the first two factors from the Netflix data matrix factorization. Movies are placed according to their factor vectors. Someone familiar with the movies shown can see clear meaning in the latent factors. The first factor vector (x-axis) has on one side lowbrow comedies and horror movies, aimed at a male or adolescent audience (*Half Baked*, *Freddy vs. Jason*), while the other side contains drama or comedy with serious undertones and strong female leads (*Sophie's Choice*, *Moonstruck*). The second factorization axis (y-axis) has independent, critically acclaimed, quirky films (*Punch-Drunk Love*, *I Heart Huckabees*) on the top, and on the bottom, mainstream formulaic films (*Armageddon*, *Runaway Bride*). There are interesting intersections



Approaches of Collaborative Filtering: Matrix Factorization



$$\min_{q^*,p^*} \sum_{(u,i)\in\kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\| q_i \|^2 + \| p_u \|^2)$$

$$\min_{p^*,q^*,b^*} \sum_{(u,i)\in\kappa} (r_{ui} - \mu - b_u - b_i - p_u^T q_i)^2 + \lambda$$
$$(\| p_u \|^2 + \| q_i \|^2 + b_u^2 + b_i^2)$$

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + q_i^T p_u(t)$$

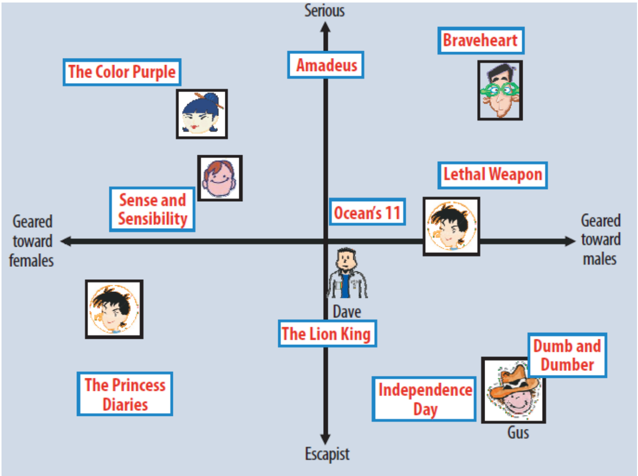
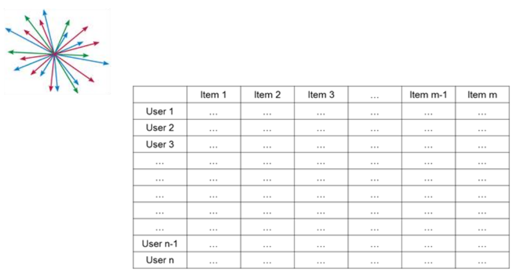
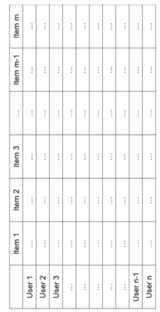


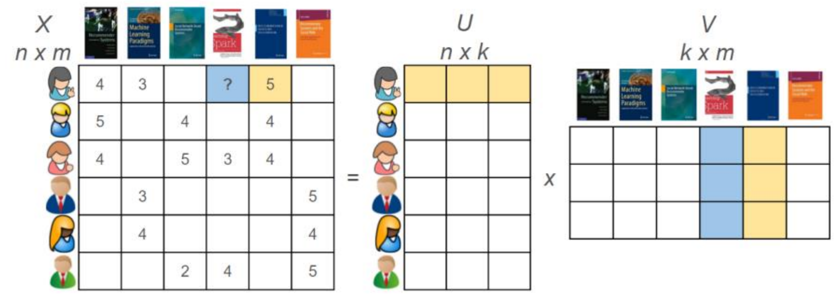
Figure 2. A simplified illustration of the latent factor approach, which characterizes both users and movies using two axes—male versus female and serious versus escapist.

Image source: https://ieeexplore.ieee.org/document/5197422

Approaches of Collaborative Filtering: Similarity Matrix Factorization





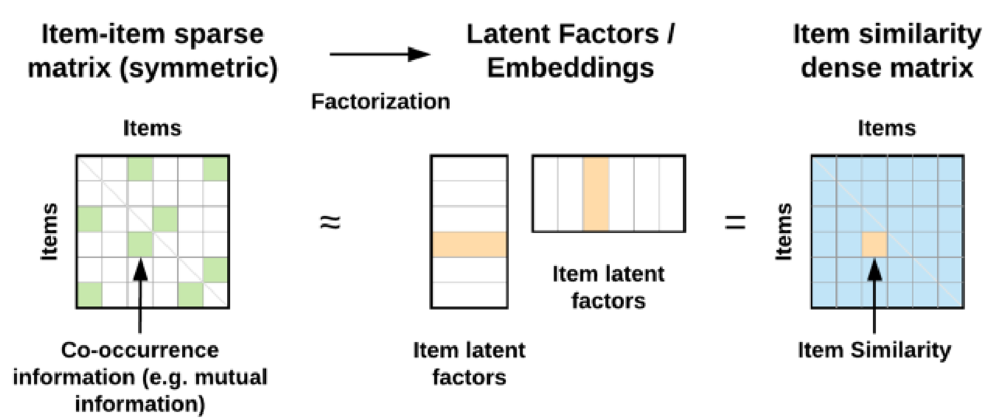


- Euclidean distance
- Cosine similarity
- Jaccard coefficient
- Pearson correlation coefficient
- …

$$\mathrm{Sim}(u_a, u_b) = \frac{\sum_{i=1}^{i'} (r_{u_{a,i}})(r_{u_{b,i}})}{\sqrt{\sum_{i=1}^{n} (r_{u_{a,i}})}\sqrt{\sum_{i=1}^{n} (r_{u_{b,i}})}}$$

$$\mathrm{Sim}(u_a, u_b) = \frac{|I_{u_a}| \cap |I_{u_b}|}{|I_{u_a}| \cup |I_{u_b}|}$$

$$\mathrm{Sim}(u_a, u_b) = \frac{\sum_{i=1}^{i'} (r_{u_{a,i}} - r_{\overline{u_a}})(r_{u_{b,i}} - r_{\overline{u_b}})}{\sqrt{\sum_{i=1}^{i'} (r_{u_{a,i}} - r_{\overline{u_a}})^2} \cdot \sqrt{\sum_{i=1}^{i'} (r_{\overline{u_{b,i}}} - r_{\overline{u_b}})^2}}$$



Evaluation of Recommender Systems: Measures

Divide by the total number of data points

Predicted output value

Actual output value

$$MAE = \boxed{\frac{1}{n}} \sum \underbrace{\left| \boxed{y} - \boxed{\hat{y}} \right|}_{\text{The absolute value of the residual}}$$

Sum of

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

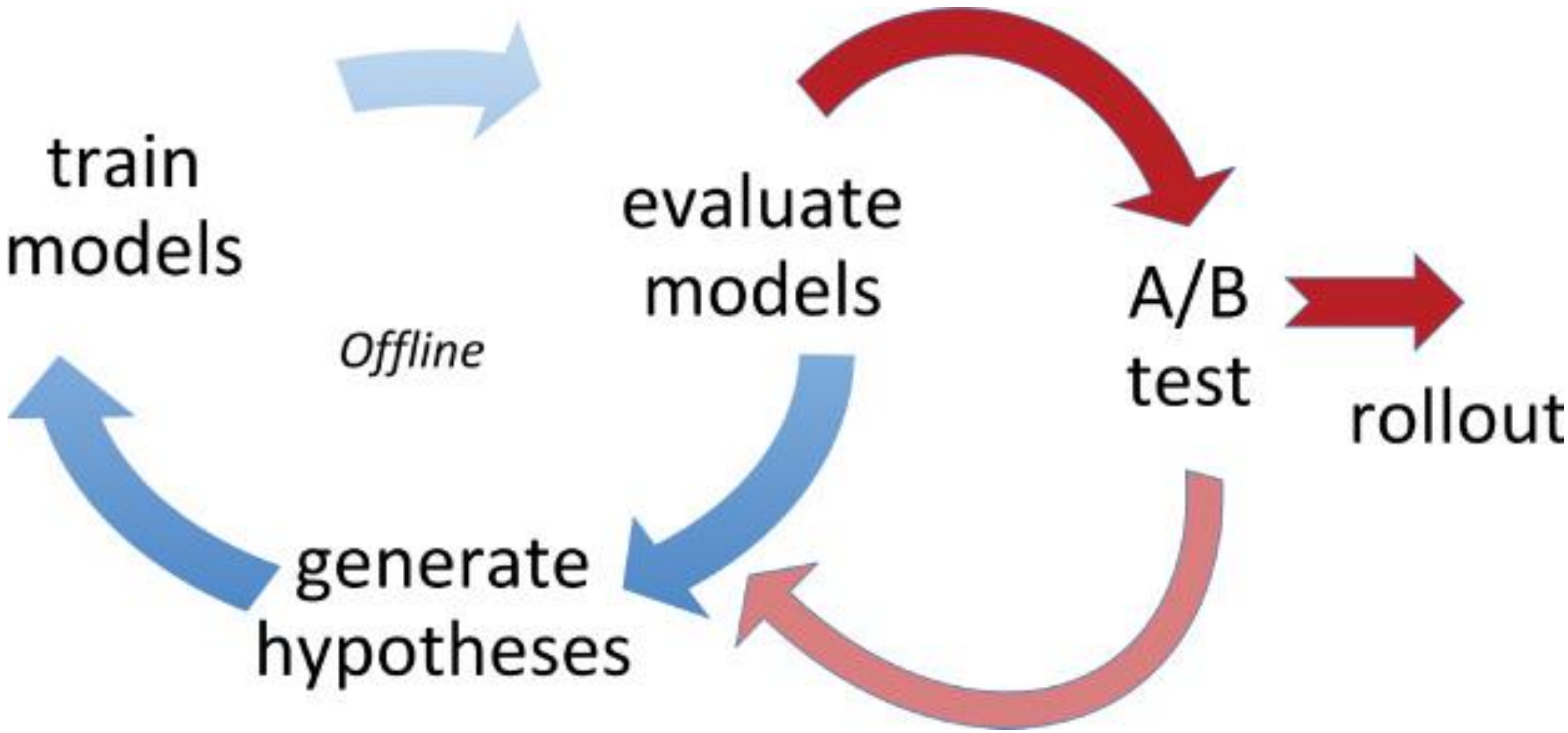
		Predicted Class	
		Positive (PP)	**Negative (PN)**
Actual Class	**Positive (AP)**	True Positive (TP)	False Negative (FN)
	Negative (AN)	False Positive (FP)	True Negative (TN)

$$\text{Diversity} = 1 - \text{Similarity}$$

CUSTOMER CHURN

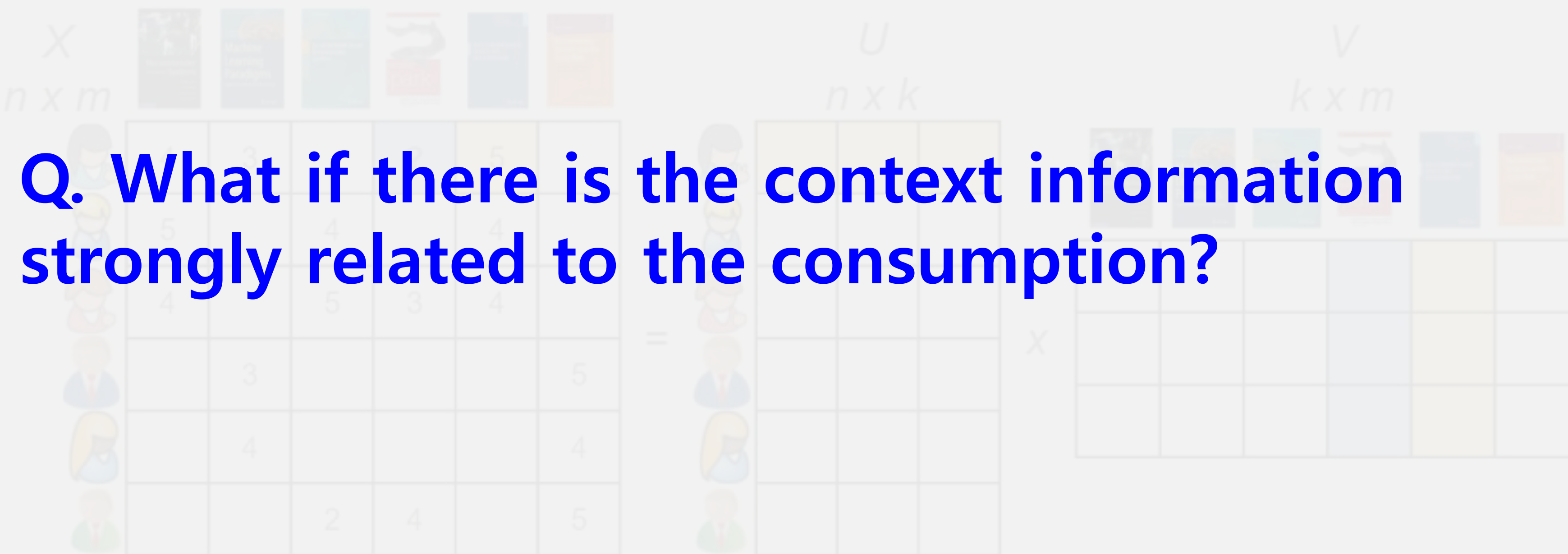
Image source: https://medium.com/fnplus/evaluating-recommender-systems-with-python-code-ae0c370c90be, https://towardsdatascience.com/evaluating-recommendation-engines-74daebbe46db

Evaluation of Recommender Systems: Verification vs. Validation



Use of the Bag-of-Words-Form Transactions Matrix

Approaches of Recommender Systems



Q. What if there is the context information strongly related to the consumption?

Approaches of Recommender Systems: A Data Perspective

- Recommender systems use analytic techniques to compute the value that a user will purchase one of the items; the techniques vary according to the purposes and data

Matrix Factorization Training Data

	i_1	i_2	i_3
u_1	2	4	
u_2		1	
u_3	3		5

vs.

Factorization Machine Training Data

	u_1	u_2	u_3	i_1	i_2	i_3	a_1	a_2	y
x_1	1	0	0	1	0	0	2.0	0.0	2
x_2	1	0	0	0	1	0	1.5	0.5	4
x_3	0	1	0	0	1	0	0.0	1.0	1
x_4	0	0	1	1	0	0	0.3	0.7	3
x_5	0	0	1	0	0	1	3.2	1.7	5

Users (u_1–u_3) | Items (i_1–i_3) | Auxiliary Features (a_1–a_2) | Observed Ratings (y)

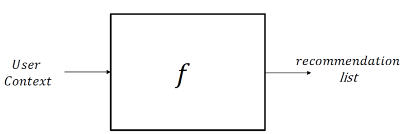
Image reference: https://towardsdatascience.com/factorization-machines-for-item-recommendation-with-implicit-feedback-data-5655a7c749db

Approaches of Recommender Systems: A Data Perspective

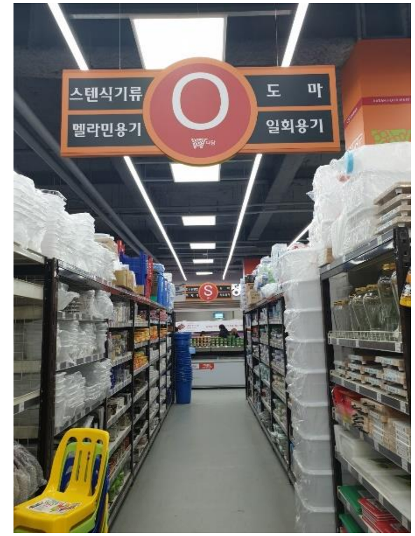
- Recommender systems use analytic techniques to compute the value that a user will purchase one of the items; the techniques vary according to the purposes and data

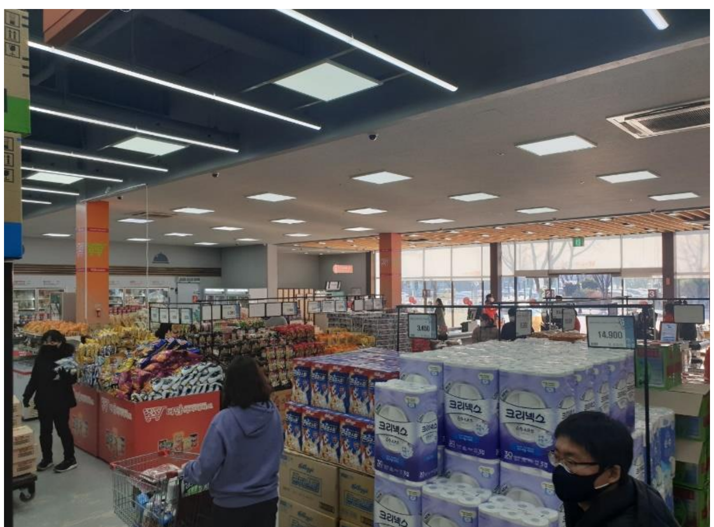
	Feature 1	Feature 2	Feature 3	…	Feature m-1	Feature m
Transaction 1	…	…	…	…	…	…
Transaction 2	…	…	…	…	…	…
Transaction 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
Transaction n-1	…	…	…	…	…	…
Transaction n	…	…	…	…	…	…

$$y = f(x)$$

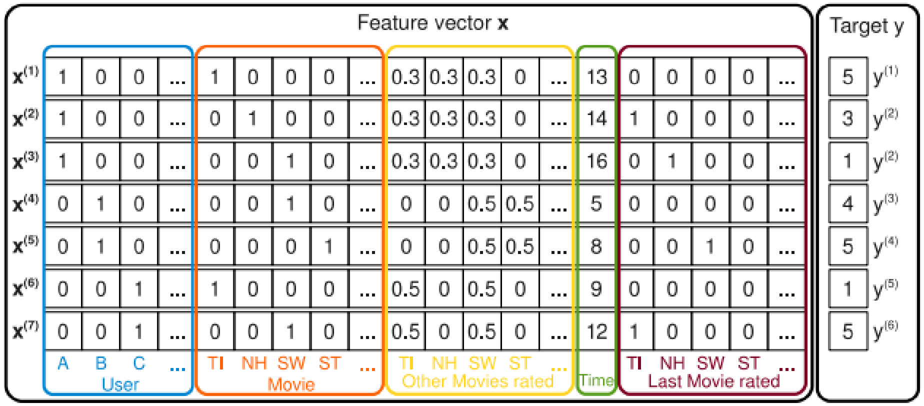


Approaches of Recommender Systems: Offline Context





Factorization Machine



	User A	User B	User C	User ...	Movie TI	Movie NH	Movie SW	Movie ST	Movie ...	Other Movies rated TI	Other Movies rated NH	Other Movies rated SW	Other Movies rated ST	Other Movies rated ...	Time	Last Movie rated TI	Last Movie rated NH	Last Movie rated SW	Last Movie rated ST	Last Movie rated ...	Target y	
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j \qquad (1)$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \qquad (2)$$

And $\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size k:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \qquad (3)$$

A row $\mathbf{v}_i$ within $\mathbf{V}$ describes the i-th variable with k factors. $k \in \mathbb{N}_0^+$ is a hyperparameter that defines the dimensionality of the factorization.

Reference: https://www.jefkine.com/recsys/2017/03/27/factorization-machines/
https://ieeexplore.ieee.org/abstract/document/5694074?casa_token=CV0m3FJ7U3UAAAAA:1rubmy3hLA6hvfRzAxZV4ykDx4kufMqEsGPX69_eJckG3BP05EJCh7DEkUcSmGQzc6JhEf37dX4

Factorization Machine

Feature vector $\mathbf{x}$	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j \tag{1}$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \tag{2}$$

And $\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size k:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \tag{3}$$

A row $\mathbf{v}_i$ within $\mathbf{V}$ describes the i-th variable with k factors. $k \in \mathbb{N}_0^+$ is a hyperparameter that defines the dimensionality of the factorization.

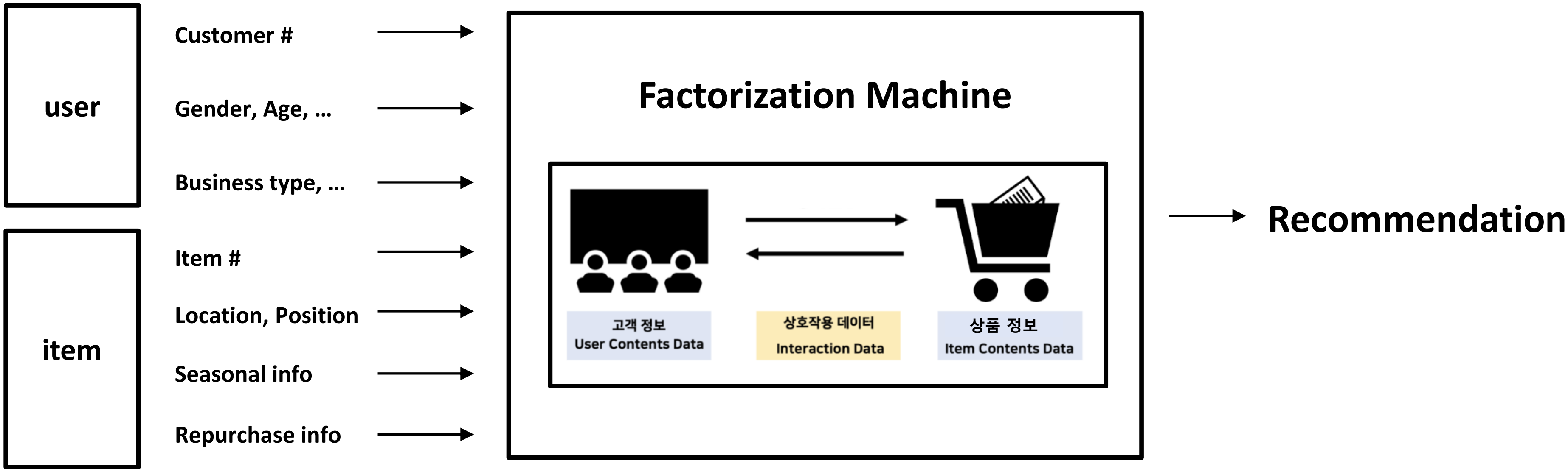
$$\begin{aligned}
&\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j \\
=&\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j - \frac{1}{2} \sum_{i=1}^{n} \langle \mathbf{v}_i, \mathbf{v}_i \rangle\, x_i\, x_i \\
=&\frac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i\, x_j - \sum_{i=1}^{n} \sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i\, x_i \right) \\
=&\frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i \right) \left(\sum_{j=1}^{n} v_{j,f}\, x_j \right) - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2 \right) \\
=&\frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2 \right)
\end{aligned}$$

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i\, x_i + \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2 \right)$$

$$\frac{\partial}{\partial \theta} \hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases}$$

Reference: https://www.jefkine.com/recsys/2017/03/27/factorization-machines/
https://ieeexplore.ieee.org/abstract/document/5694074?casa_token=CV0m3FJ7U3UAAAAA:1rubmy3hLA6hvfRzAxZV4ykDx4kufMqEsGPX69_eJckG3BP05EJCh7DEkUcSmGQzc6JhEf37dX4

Factorization Machine for Offline Contexts



Factorization Machine Training Data

	u_1	u_2	u_3	i_1	i_2	i_3	a_1	a_2	y
x_1	1	0	0	1	0	0	2.0	0.0	2
x_2	1	0	0	0	1	0	1.5	0.5	4
x_3	0	1	0	0	1	0	0.0	1.0	1
x_4	0	0	1	1	0	0	0.3	0.7	3
x_5	0	0	1	0	0	1	3.2	1.7	5
	Users			Items			Auxiliary Features		Observed Ratings

User Info
- (사업자 회원) 업종유형

상품의 매장 내 구역 및 위치
- 상품 매대 정보
- 상품 통로 정보

구매 관련
- 구매 시즌
- 과거에 구매했던 상품 → 재구매 가중치

Feature Vector

	사업자회원번호				상품 코드					구매 횟수	상품 위치					사업자 업종 유형					Target y	
$x^{(1)}$	1	0	0	...	1	0	0	0	...	1.58	0	0	0	0	...	0	0	0	0	...	5	$y^{(1)}$
$x^{(2)}$	1	0	0	...	0	1	0	0	...	1.00	1	0	0	0	...	1	0	0	0	...	3	$y^{(2)}$
$x^{(3)}$	1	0	0	...	0	0	1	0	...	2.81	0	1	0	0	...	0	1	0	0	...	1	$y^{(2)}$
$x^{(4)}$	0	1	0	...	0	0	1	0	...	1.00	0	0	0	0	...	0	0	0	0	...	4	$y^{(3)}$
$x^{(5)}$	0	1	0	...	0	0	0	1	...	1.58	0	0	1	0	...	0	0	1	0	...	5	$y^{(4)}$
$x^{(6)}$	0	0	1	...	1	0	0	0	...	1.85	0	0	0	0	...	0	0	0	0	...	1	$y^{(5)}$
$x^{(7)}$	0	0	1	...	0	0	1	0	...	2.81	1	0	0	0	...	1	0	0	0	...	5	$y^{(6)}$

Assignment 2 (by 9.16 11:59 pm)

- There are two types of practice demonstrated by TAs. One approach is to use the user-item matrix and the other is to use the transaction-feature matrix. By yourself, (1) identify several (i.e., top k) recommendations using the two approaches with the provided datasets.
- (2) In each practice, evaluate and interpret the recommendation outcomes quantitatively (e.g., calculate the recall, calculate the similarities between the recommended items) and qualitatively (e.g., interpret the factorization outcome, identify the characteristics of the top k recommended items). Do it all by yourself, and describe the analysis/interpretation process and outcome in detail.
- (3) Assume you need to use your recommender system for real-world service (i.e., streaming service or hypermarket service). How can you improve your recommender system to be used for the service effectively? For example, what kinds of data should you use further? How would you design a method for using/learning the data? Think beyond these examples in your own creative, unique way!
- (4) You must have your own interested or favorite service WITHOUT a recommender system (i.e., it should be different with the intelligent services discussed in the class). Discuss the requirements of original recommender system development for the service. Describe the requirements in detail.
- (5) If you would conduct a study on the recommender system development for the service, how would you conduct the research in your own creative, unique way? What kinds of data and methods are you going to collect, analyze, and learn? Describe your service intelligence development plan in detail. If possible, visualize your plan clearly (e.g., draw an image, construct a mathematical model). To facilitate your thinking, you may want to identify and review a recommender system paper related to the service you are interested or concerned.
- Upload your code and a several paragraph essay on the tasks (1)~(5) in the Blackboard.

Matrix Factorization Practice

Demonstrated by TA Seo

Factorization Machine Practice

Demonstrated by TA Shin

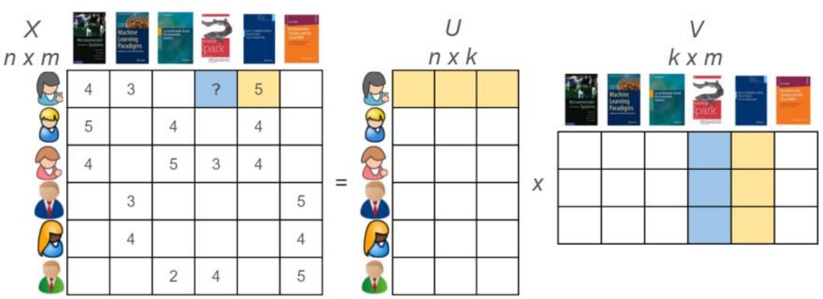
Assignment 2 (by 9.16 11:59 pm)

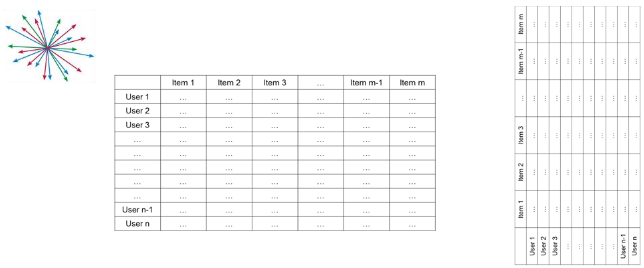
- There are two types of practice demonstrated by TAs. One approach is to use the user-item matrix and the other is to use the transaction-feature matrix. By yourself, (1) identify several (i.e., top k) recommendations using the two approaches with the provided datasets.
- (2) In each practice, evaluate and interpret the recommendation outcomes quantitatively (e.g., calculate the recall, calculate the similarities between the recommended items) and qualitatively (e.g., interpret the factorization outcome, identify the characteristics of the top k recommended items). Do it all by yourself, and describe the analysis/interpretation process and outcome in detail.
- (3) Assume you need to use your recommender system for real-world service (i.e., streaming service or hypermarket service). How can you improve your recommender system to be used for the service effectively? For example, what kinds of data should you use further? How would you design a method for using/learning the data? Think beyond these examples in your own creative, unique way!
- (4) You must have your own interested or favorite service WITHOUT a recommender system (i.e., it should be different with the intelligent services discussed in the class). Discuss the requirements of original recommender system development for the service. Describe the requirements in detail.
- (5) If you would conduct a study on the recommender system development for the service, how would you conduct the research in your own creative, unique way? What kinds of data and methods are you going to collect, analyze, and learn? Describe your service intelligence development plan in detail. If possible, visualize your plan clearly (e.g., draw an image, construct a mathematical model). To facilitate your thinking, you may want to identify and review a recommender system paper related to the service you are interested or concerned.
- Upload your code and a several paragraph essay on the tasks (1)~(5) in the Blackboard.

Concluding Remarks

Approaches of Collaborative Filtering: Emergence of Deep Learning

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data
  - Collaborative filtering
    - Uses an information filtering technique based on the user's previous evaluation of items or history of previous purchases





- Euclidean distance
- Cosine similarity
- Jaccard coefficient
- Pearson correlation coefficient
- …

$$\mathrm{Sim}(u_a, u_b) = \frac{\sum_{i=1}^{i'} (r_{u_{a,i}})(r_{u_{b,i}})}{\sqrt{\sum_{i=1}^{n} (r_{u_{a,i}})}\sqrt{\sum_{i=1}^{n} (r_{u_{b,i}})}}$$

$$\mathrm{Sim}(u_a, u_b) = \frac{|I_{u_a}| \cap |I_{u_b}|}{|I_{u_a}| \cup |I_{u_b}|}$$

$$\mathrm{Sim}(u_a, u_b) = \frac{\sum_{i=1}^{i'} (r_{u_{a,i}} - r_{\overline{u_a}})(r_{u_{b,i}} - r_{\overline{u_b}})}{\sqrt{\sum_{i=1}^{i'} (r_{u_{a,i}} - r_{\overline{u_a}})^2} \cdot \sqrt{\sum_{i=1}^{i'} (r_{\overline{u_{b,i}}} - r_{\overline{u_b}})^2}}$$

Approaches of Collaborative Filtering: Emergence of Deep Learning

- Deep-learning-based nonlinearity consideration complements the traditional approaches

Neural Collaborative Filtering*

Xiangnan He
National University of Singapore, Singapore
xiangnanhe@gmail.com

Lizi Liao
National University of Singapore, Singapore
liaolizi.llz@gmail.com

Hanwang Zhang
Columbia University
USA
hanwangzhang@gmail.com

Liqiang Nie
Shandong University
China
nieliqiang@gmail.com

Xia Hu
Texas A&M University
USA
hu@cse.tamu.edu

Tat-Seng Chua
National University of Singapore, Singapore
dcscts@nus.edu.sg

ABSTRACT

In recent years, deep neural networks have yielded immense success on speech recognition, computer vision and natural language processing. However, the exploration of deep neural networks on recommender systems has received relatively less scrutiny. In this work, we strive to develop techniques based on neural networks to tackle the key problem in recommendation — collaborative filtering — on the basis of implicit feedback.

Although some recent work has employed deep learning for recommendation, they primarily used it to model auxiliary information, such as textual descriptions of items and acoustic features of musics. When it comes to model the key factor in collaborative filtering — the interaction between user and item features, they still resorted to matrix factorization and applied an inner product on the latent features of users and items.

By replacing the inner product with a neural architecture that can learn an arbitrary function from data, we present a general framework named NCF, short for *Neural network-based Collaborative Filtering*. NCF is generic and can express and generalize matrix factorization under its framework. To supercharge NCF modelling with non-linearities, we propose to leverage a multi-layer perceptron to learn the user–item interaction function. Extensive experiments on two real-world datasets show significant improvements of our proposed NCF framework over the state-of-the-art methods. Empirical evidence shows that using deeper layers of neural networks offers better recommendation performance.

Keywords

Collaborative Filtering, Neural Networks, Deep Learning, Matrix Factorization, Implicit Feedback

*NExT research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052569

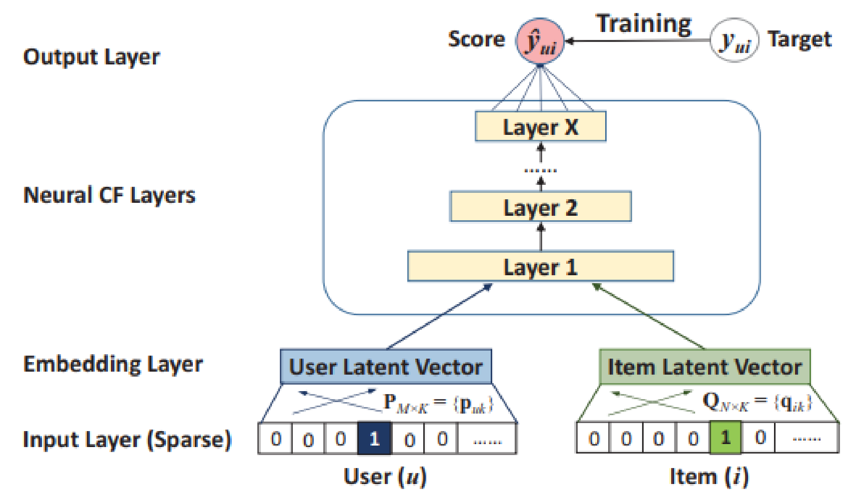
1. INTRODUCTION

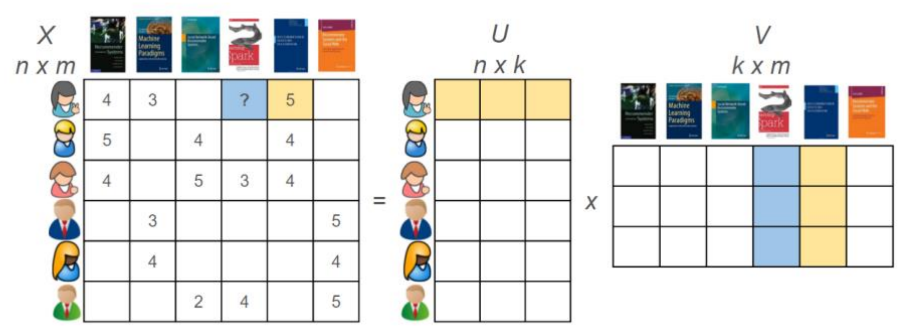
In the era of information explosion, recommender systems play a pivotal role in alleviating information overload, having been widely adopted by many online services, including E-commerce, online news and social media sites. The key to a personalized recommender system is in modelling users' preference on items based on their past interactions (*e.g.*, ratings and clicks), known as collaborative filtering [31, 46]. Among the various collaborative filtering techniques, matrix factorization (MF) [14, 21] is the most popular one, which projects users and items into a shared latent space, using a vector of latent features to represent a user or an item. Thereafter a user's interaction on an item is modelled as the inner product of their latent vectors.

Popularized by the Netflix Prize, MF has become the *de facto* approach to latent factor model-based recommendation. Much research effort has been devoted to enhancing MF, such as integrating it with neighbor-based models [21], combining it with topic models of item content [38], and extending it to factorization machines [26] for a generic modelling of features. Despite the effectiveness of MF for collaborative filtering, it is well-known that its performance can be hindered by the simple choice of the interaction function — inner product. For example, for the task of rating prediction on explicit feedback, it is well known that the performance of the MF model can be improved by incorporating user and item bias terms into the interaction function[1]. While it seems to be just a trivial tweak for the inner product operator [14], it points to the positive effect of designing a better, dedicated interaction function for modelling the latent feature interactions between users and items. The inner product, which simply combines the multiplication of latent features linearly, may not be sufficient to capture the complex structure of user interaction data.

This paper explores the use of deep neural networks for learning the interaction function from data, rather than a handcraft that has been done by many previous work [18, 21]. The neural network has been proven to be capable of approximating any continuous function [17], and more recently deep neural networks (DNNs) have been found to be effective in several domains, ranging from computer vision, speech recognition, to text processing [5, 10, 15, 47]. However, there is relatively little work on employing DNNs for recommendation in contrast to the vast amount of literature

[1] http://alex.smola.org/teaching/berkeley2012/slides/8_Recommender.pdf





Approaches of Collaborative Filtering: Emergence of Deep Learning

- Deep-learning-based abstraction/representation complements the traditional approaches

ITEM2VEC: NEURAL ITEM EMBEDDING FOR COLLABORATIVE FILTERING

*Oren Barkan^* and Noam Koenigstein**

^Tel Aviv University
*Microsoft

ABSTRACT

Many Collaborative Filtering (CF) algorithms are item-based in the sense that they analyze item-item relations in order to produce item similarities. Recently, several works in the field of Natural Language Processing (NLP) suggested to learn a latent representation of words using neural embedding algorithms. Among them, the Skip-gram with Negative Sampling (SGNS), also known as word2vec, was shown to provide state-of-the-art results on various linguistics tasks. In this paper, we show that item-based CF can be cast in the same framework of neural word embedding. Inspired by SGNS, we describe a method we name item2vec for item-based CF that produces embedding for items in a latent space. The method is capable of inferring item-item relations even when user information is not available. We present experimental results that demonstrate the effectiveness of the item2vec method and show it is competitive with SVD.

Index terms – skip-gram, word2vec, neural word embedding, collaborative filtering, item similarity, recommender systems, market basket analysis, item-item collaborative filtering, item recommendations.

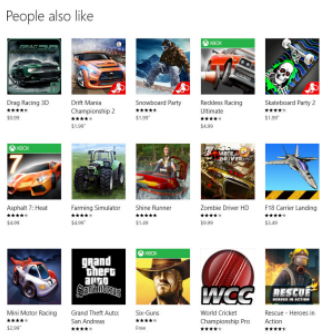


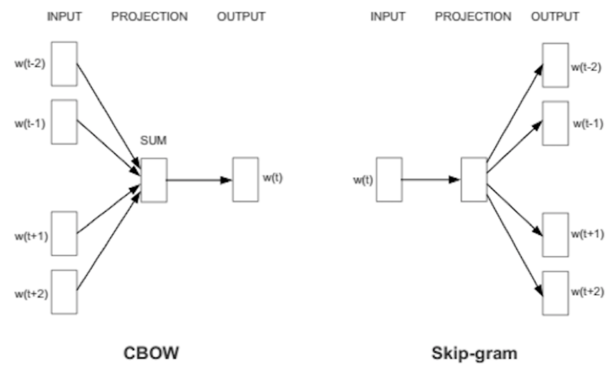
Fig. 1. Recommendations in Windows 10 Store based on similar items to Need For Speed.

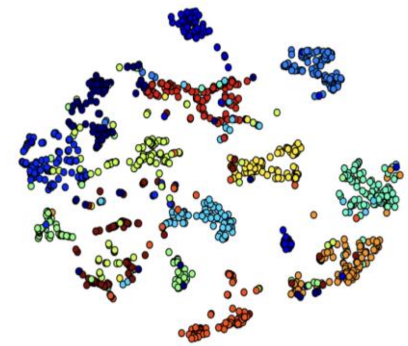
1. INTRODUCTION AND RELATED WORK

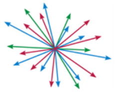
Computing item similarities is a key building block in modern recommender systems. While many recommendation algorithms are focused on learning a low dimensional embedding of users and items simultaneously [1, 2, 3], computing item similarities is an end in itself. Item similarities are extensively used by online retailers for many different recommendation tasks. This paper deals with the overlooked task of learning item similarities by embedding items in a low dimensional space.

Item-based similarities are used by online retailers for recommendations based on a single item. For example, in the Windows 10 App Store, the details page of each app or game includes a list of other similar apps titled "People also like". This list can be extended to a full page recommendation list of items similar to the original app as shown in Fig. 1. Similar recommendation lists which are based merely on similarities to a single item exist in most online stores e.g., Amazon, Netflix, Google Play, iTunes store and many others.

The single item recommendations are different than the more "traditional" user-to-item recommendations because they are usually shown in the context of an explicit user interest in a specific item and in the context of an explicit user intent to purchase. Therefore, single item recommendations based on item similarities often have higher Click-Through Rates (CTR) than user-to-item recommendations and consequently responsible for a larger share of sales or revenue.





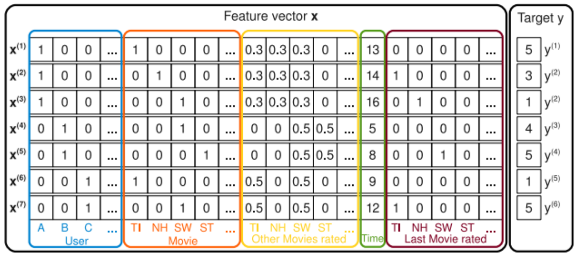


	Item 1	Item 2	Item 3	…	Item m-1	Item m
User 1	…	…	…	…	…	…
User 2	…	…	…	…	…	…
User 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
User n-1	…	…	…	…	…	…
User n	…	…	…	…	…	…

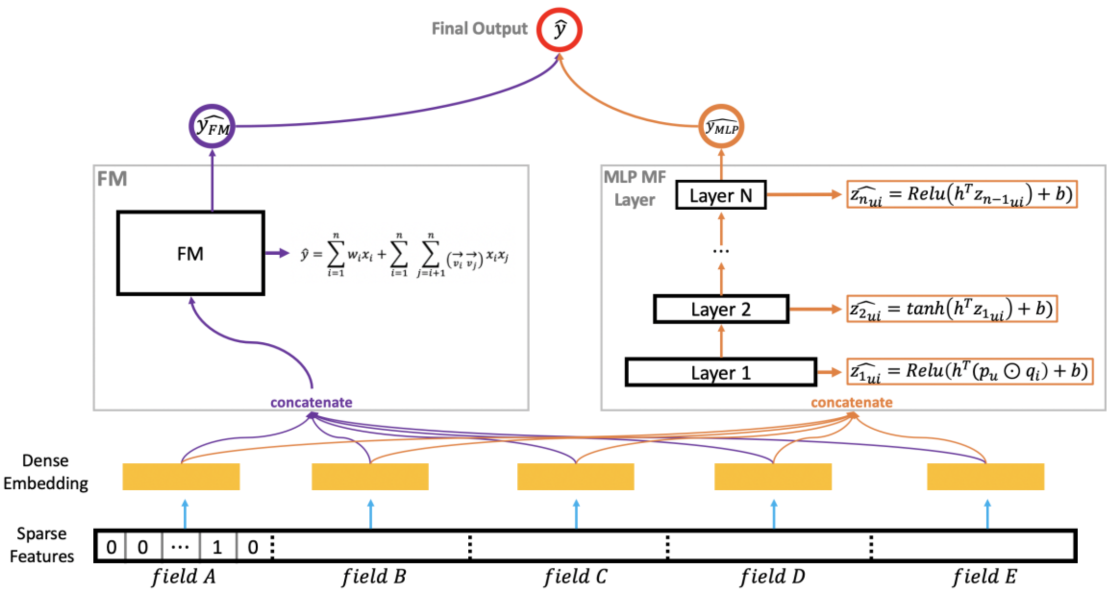
- Euclidean distance
- Cosine similarity
- Jaccard coefficient
- Pearson correlation coefficient
- …

Approaches of Collaborative Filtering: Emergence of Deep Learning

- Deep-learning-based nonlinearity consideration complements the traditional approaches
- Deep-learning-based abstraction/representation complements the traditional approaches



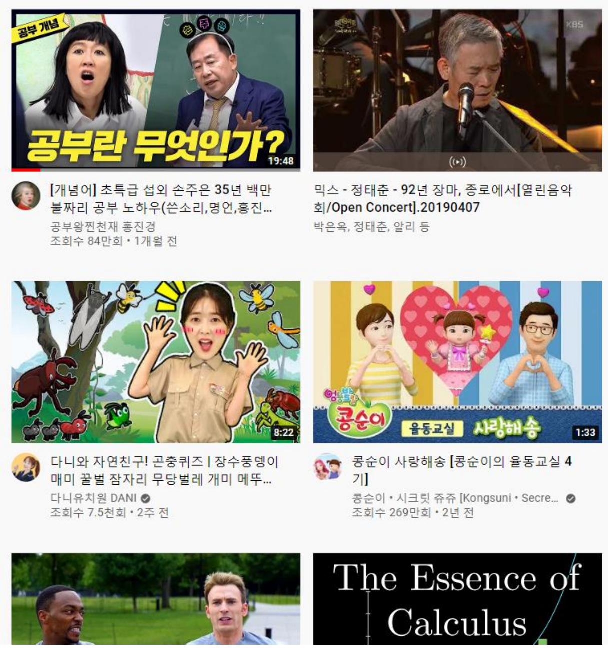
	Feature vector x																			Target y	
	User				Movie					Other Movies rated					Time	Last Movie rated					
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...		TI	NH	SW	ST	...	
$x^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5 $y^{(1)}$
$x^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3 $y^{(2)}$
$x^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1 $y^{(2)}$
$x^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4 $y^{(3)}$
$x^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5 $y^{(4)}$
$x^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1 $y^{(5)}$
$x^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5 $y^{(6)}$



Reference: https://techblog-history-younghunjo1.tistory.com/208

Approaches of Recommender Systems: A Categorization

- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data
  - Content-based filtering
  - Collaborative filtering

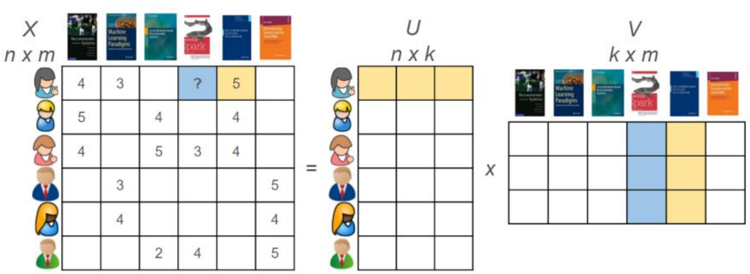


Approaches of Recommender Systems: Our Focus

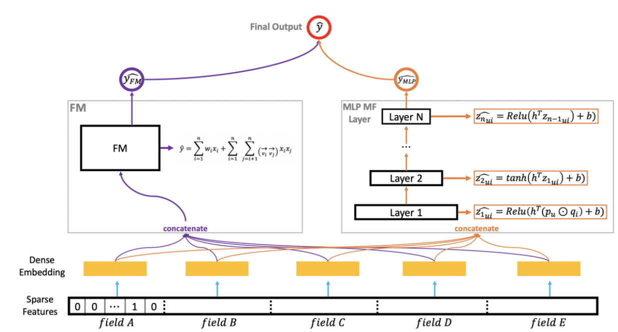
- Recommender systems use analytic techniques to compute the value that a user would purchase one of the items; the techniques vary according to the purposes and data

	Item 1	Item 2	Item 3	…	Item m-1	Item m
User 1	…	…	…	…	…	…
User 2	…	…	…	…	…	…
User 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
User n-1	…	…	…	…	…	…
User n	…	…	…	…	…	…

	Feature 1	Feature 2	Feature 3	…	Feature m-1	Feature m
Transaction 1	…	…	…	…	…	…
Transaction 2	…	…	…	…	…	…
Transaction 3	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
Transaction n-1	…	…	…	…	…	…
Transaction n	…	…	…	…	…	…



$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j$$



Discussion

- Relation and gap between the recommender system and service quality
- Considerations of recommender system development for real-world services:

  (1) data, (2) model, (3) service speed, (4) service UI, (5) model and service evaluation …
- Beyond the user-item matrix
  - Consideration of the side information (details of items and user contexts) is also required
  - What other customer variabilities should be considered in recommendation?
- Objectives of recommendation from the customer vs. operations perspectives
- Dealing with the customer's cognitive processes unknown
- Knowledge discovery (customer understanding) for recommender system development
  - Performance of model + Explainability of model + Interpretability of result
- Ethics around the recommender systems

Assignment 2 (by 9.16 11:59 pm)

- There are two types of practice demonstrated by TAs. One approach is to use the user-item matrix and the other is to use the transaction-feature matrix. By yourself, (1) identify several (i.e., top k) recommendations using the two approaches with the provided datasets.
- (2) In each practice, evaluate and interpret the recommendation outcomes quantitatively (e.g., calculate the recall, calculate the similarities between the recommended items) and qualitatively (e.g., interpret the factorization outcome, identify the characteristics of the top k recommended items). Do it all by yourself, and describe the analysis/interpretation process and outcome in detail.
- (3) Assume you need to use your recommender system for real-world service (i.e., streaming service or hypermarket service). How can you improve your recommender system to be used for the service effectively? For example, what kinds of data should you use further? How would you design a method for using/learning the data? Think beyond these examples in your own creative, unique way!
- (4) You must have your own interested or favorite service WITHOUT a recommender system (i.e., it should be different with the intelligent services discussed in the class). Discuss the requirements of original recommender system development for the service. Describe the requirements in detail.
- (5) If you would conduct a study on the recommender system development for the service, how would you conduct the research in your own creative, unique way? What kinds of data and methods are you going to collect, analyze, and learn? Describe your service intelligence development plan in detail. If possible, visualize your plan clearly (e.g., draw an image, construct a mathematical model). To facilitate your thinking, you may want to identify and review a recommender system paper related to the service you are interested or concerned.
- Upload your code and a several paragraph essay on the tasks (1)~(5) in the Blackboard.