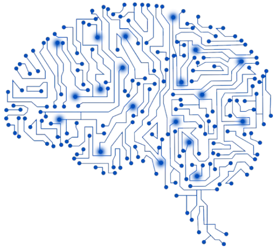
Recommendation System

Part II: Practical Recommendation Techniques

Duc-Trong Le (PhD), VNU-UET

November 2022

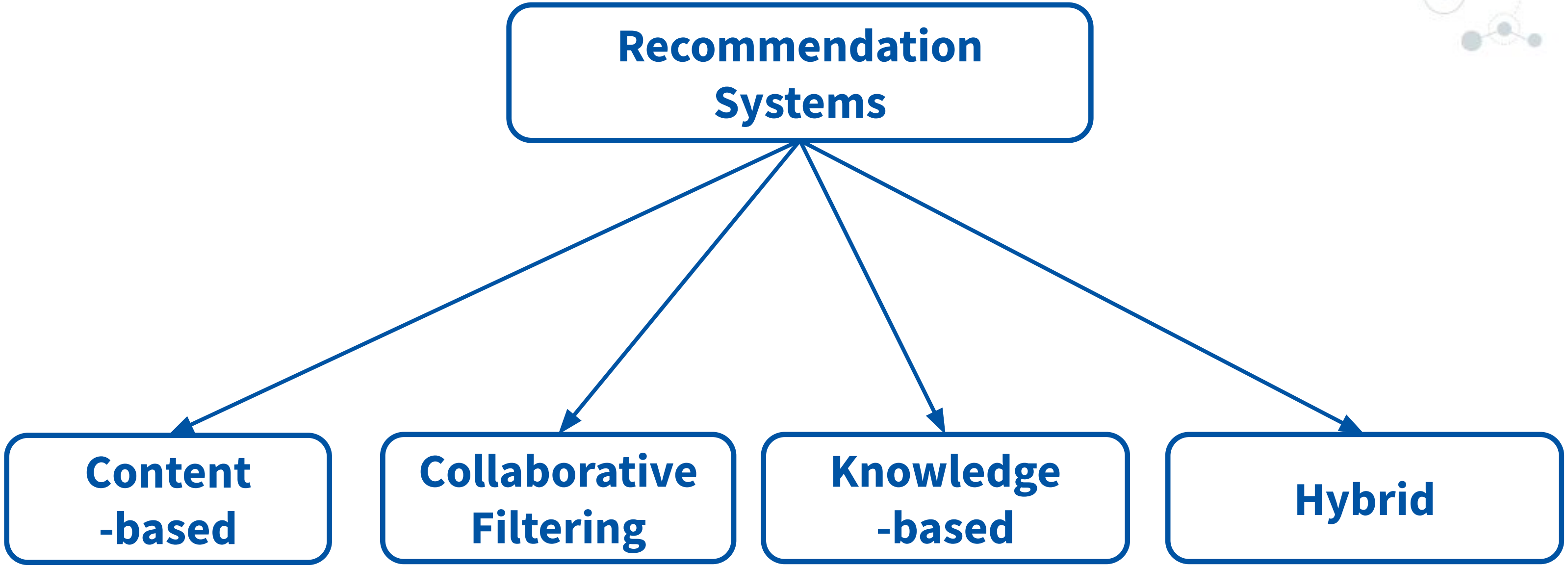


Recap: Item, User & Behavior

Customer A

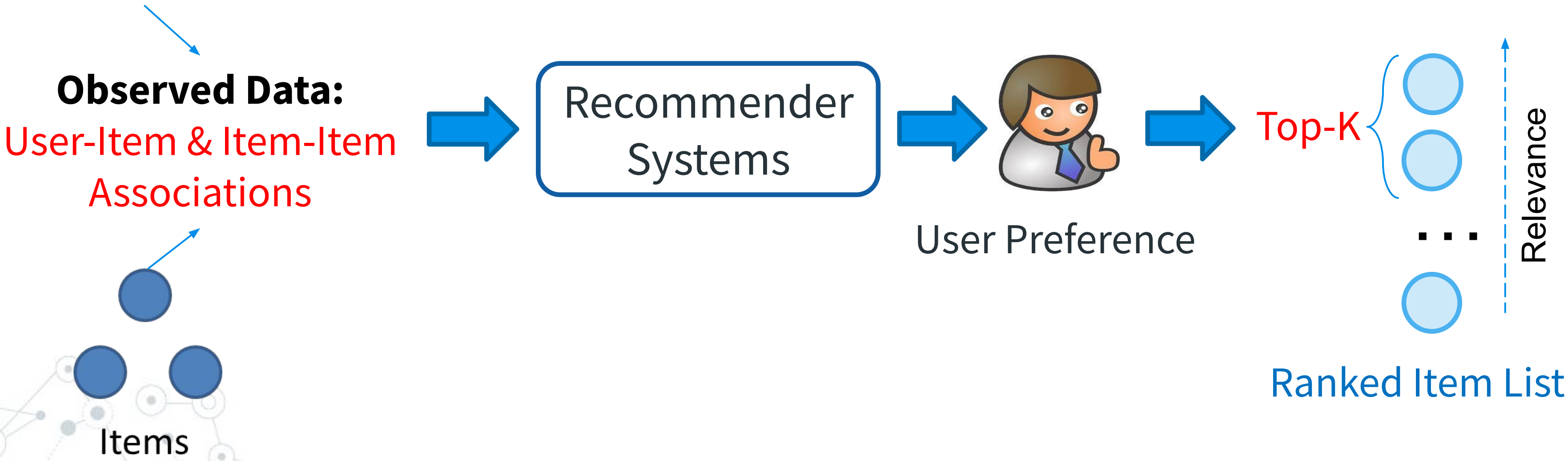
	Behavior	Item
YouTube	Click, Like/Dislike, Comment, Search	**Video**
amazon	Search, Click, Add-to-cart, Purchase, Review, Rate	**Product**
NETFLIX	Search, Click, Rate, Like	**Movie**
BáoMới.com	Search, Read, Duration, Comment	**News**
…	…	…

Recap: Types of Recommendation Systems



Recap: Top-K Recommendation

<u>Main idea</u>: Learning **user preferences** via modeling **associations** to generate **top-K recommendations**.



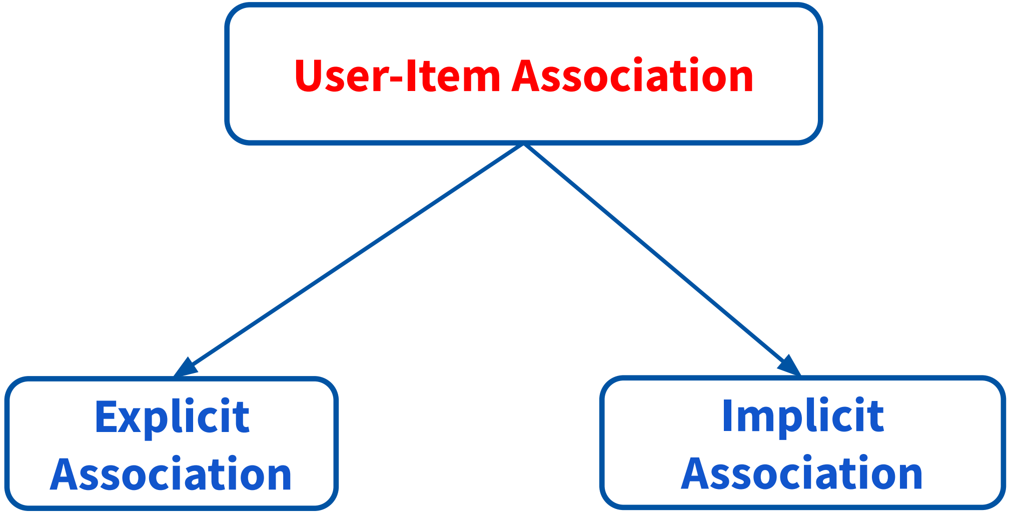
Outline

- Modeling User-Item Associations
  - Matrix Factorization & Variants
  - Bayesian Personalized Ranking & Variants
- Modeling Item-Item Associations
  - Latent Factor-based Methods
  - Neural Network-based Methods
- Challenges and Suggestions
- Real-life Recommendation Systems

1.

Modeling
User-Item Associations

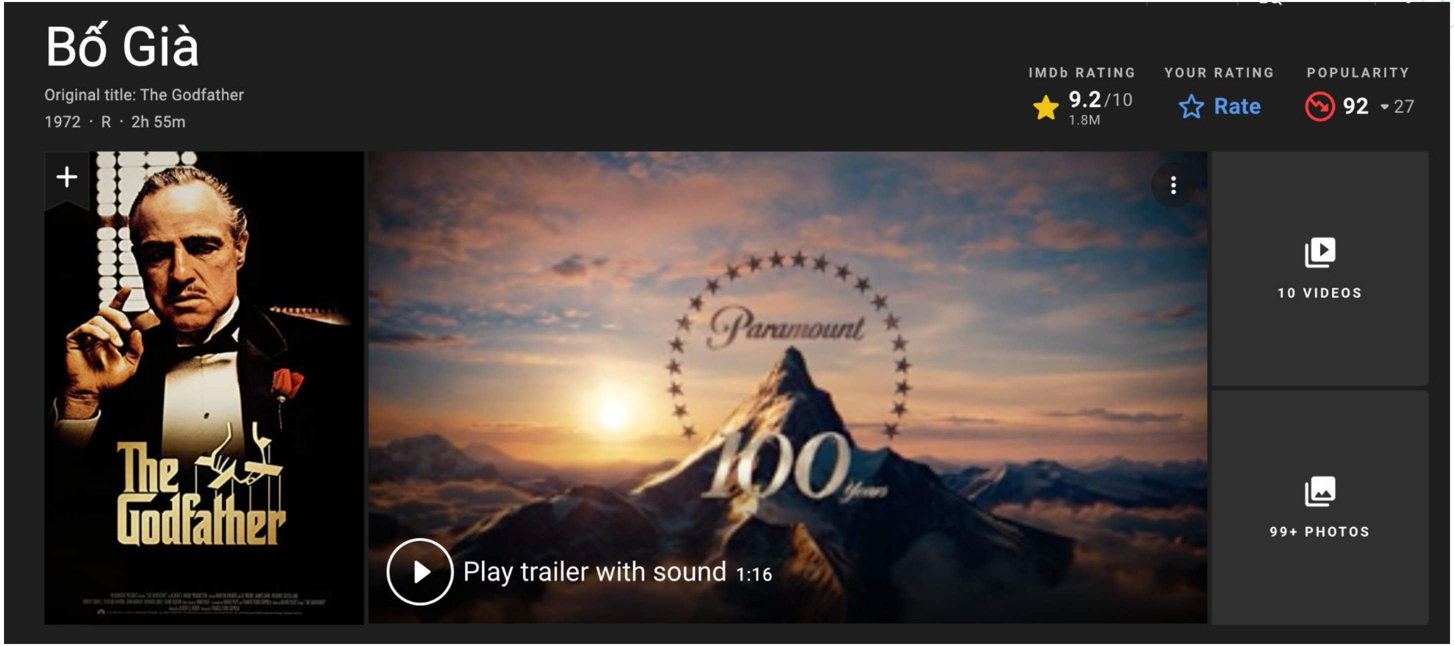
Recap: User-Item Association



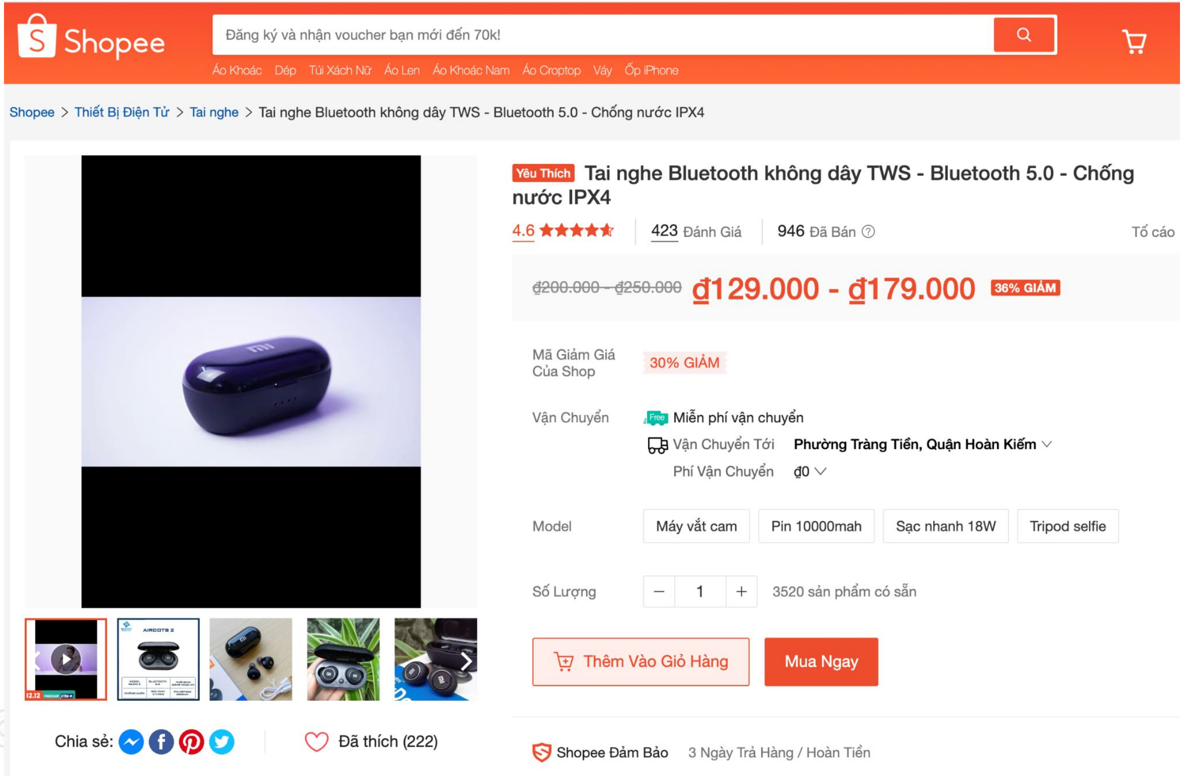
e.g., rating, score, stars

e.g., click, view, purchase

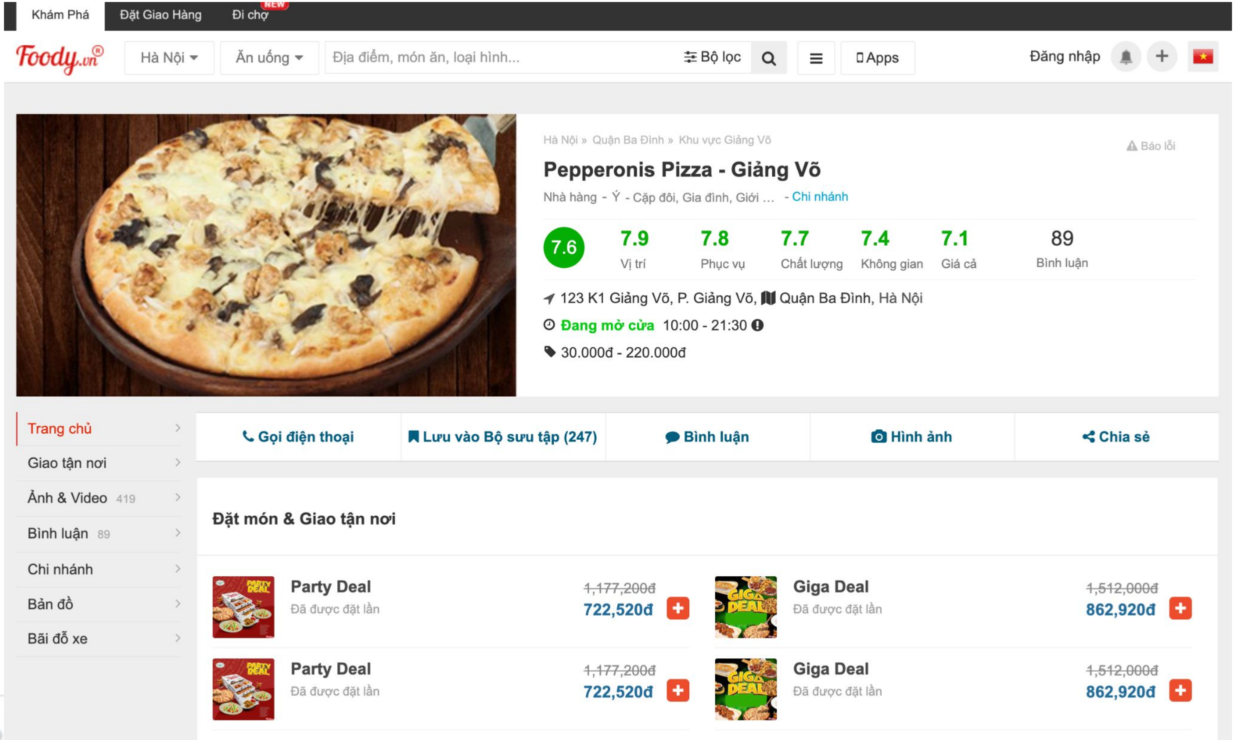
Explicit Association - IMDB



Explicit Association - Shopee



Explicit Association - Foody

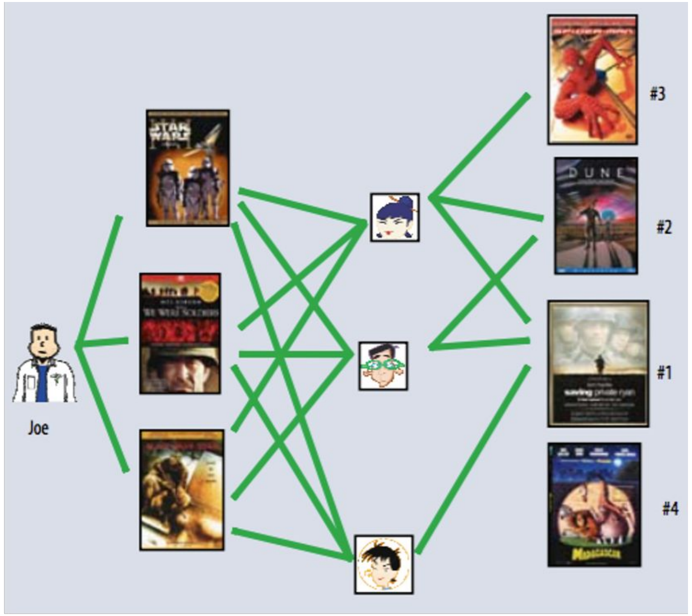


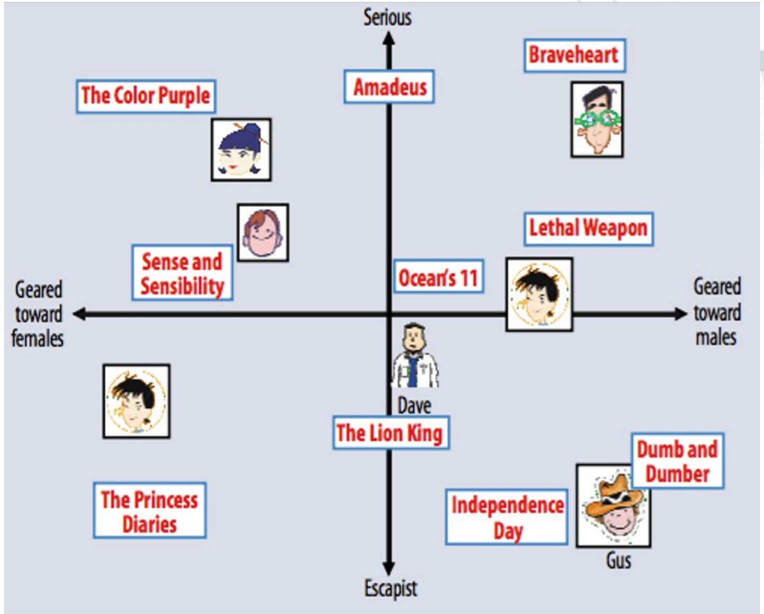
Explicit Association-based Recommendation

X $n \times m$						
	4	3		?	5	
	5		4		4	
	4		5	3	4	
		3				5
		4				4
			2	4		5

Could we estimate unknown ratings?

Two Types of Collaborative Filtering

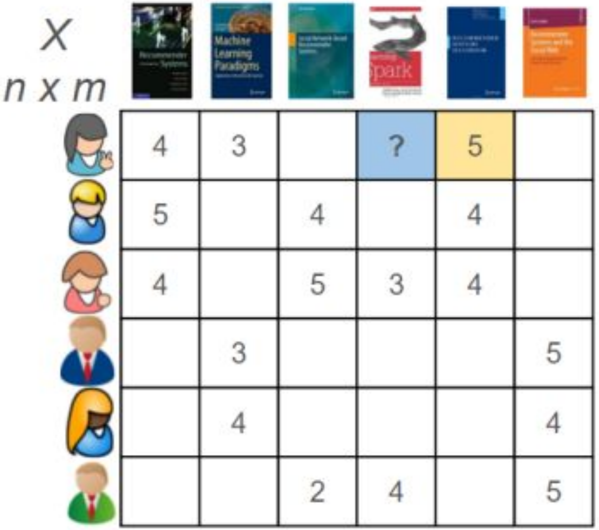




Neighborhood Method

Latent Factor Method

Neighborhood Method

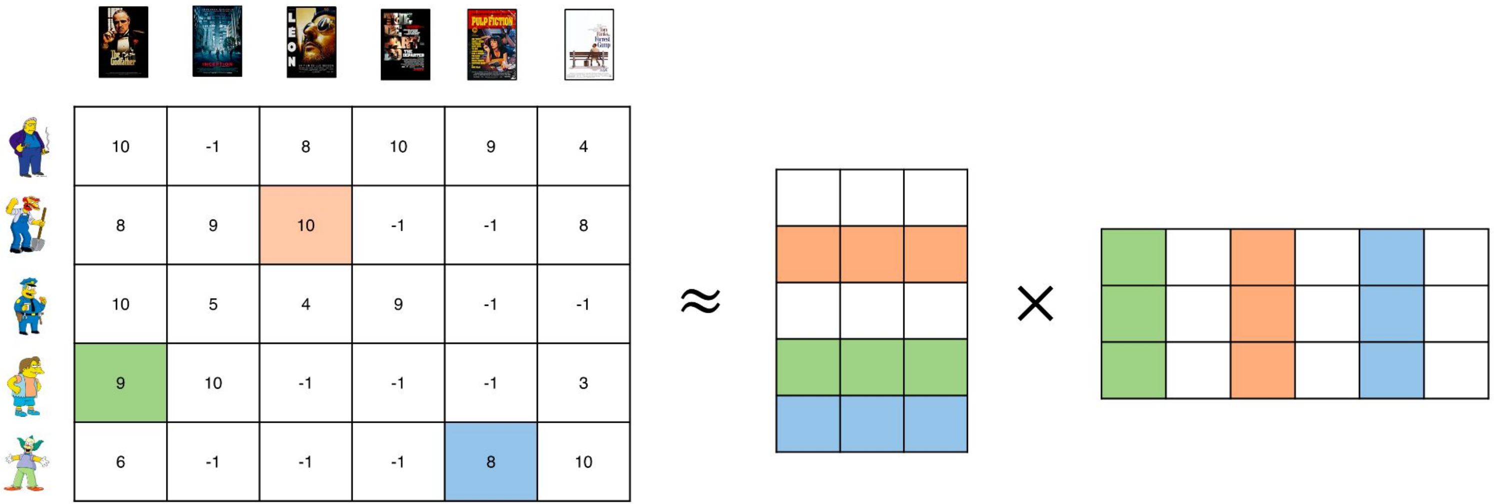


4	3		?	5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in \mathcal{N}(u,i)} \bar{y}_{i,u_j} \operatorname{sim}(u, u_j)}{\sum_{u_j \in \mathcal{N}(u,i)} |\operatorname{sim}(u, u_j)|}$$

Rating is computed via similar users/items.

Latent Factor Method - Matrix Factorization



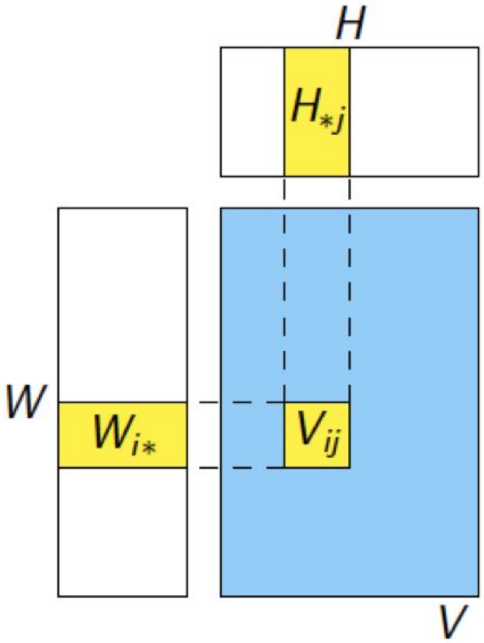
User-item Interaction Matrix

User Matrix

Item Matrix

Rating is the product of latent vectors, i.e.,
user/item latent representations.

Matrix Factorization (1)



- User vectors:

$$(W_{u*})^T \in \mathbb{R}^r$$

- Item vectors:

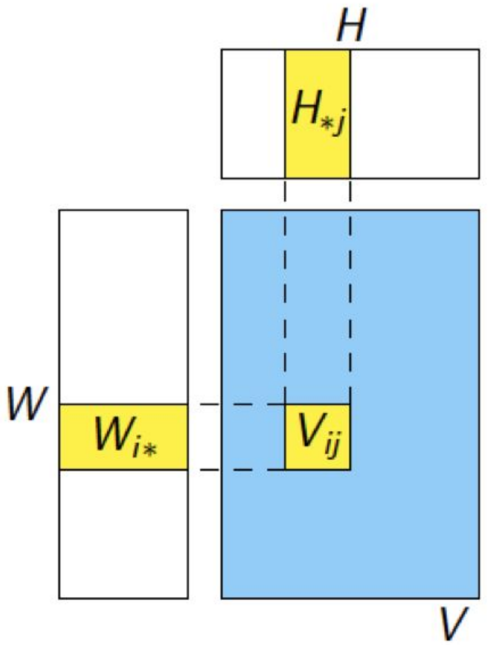
$$H_{*i} \in \mathbb{R}^r$$

- Rating prediction:

$$\begin{aligned} V_{ui} &= W_{u*} H_{*i} \\ &= [WH]_{ui} \end{aligned}$$

Ref: Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.

Matrix Factorization (2)



$$\sum_{(u,i)\in\mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2 + \lambda\left(\sum_i ||\mathbf{w}_i||^2 + \sum_u ||\mathbf{h}_u||^2\right)$$

require that the loss can be written as

$$L = \sum_{(i,j)\in Z} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$$

Algorithm 1 SGD for Matrix Factorization

Require: A training set Z, initial values $\boldsymbol{W}_0$ and $\boldsymbol{H}_0$

while not converged **do** {step}

Select a training point $(i, j) \in Z$ uniformly at random.

$\boldsymbol{W}'_{i*} \leftarrow \boldsymbol{W}_{i*} - \epsilon_n N \frac{\partial}{\partial \boldsymbol{W}_{i*}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$

$\boldsymbol{H}_{*j} \leftarrow \boldsymbol{H}_{*j} - \epsilon_n N \frac{\partial}{\partial \boldsymbol{H}_{*j}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$

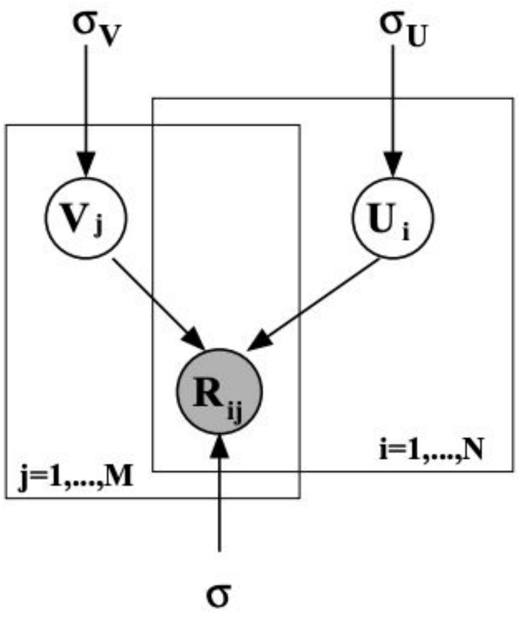
$\boldsymbol{W}_{i*} \leftarrow \boldsymbol{W}'_{i*}$

end while

step size

Ref: Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.

Probabilistic Matrix Factorization (PMF)



$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left[\mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)\right]^{I_{ij}}$$

$$p(U|\sigma_U^2) = \prod_{i=1}^{N}\mathcal{N}(U_i|0,\sigma_U^2\mathbf{I}), \qquad p(V|\sigma_V^2) = \prod_{j=1}^{M}\mathcal{N}(V_j|0,\sigma_V^2\mathbf{I}).$$

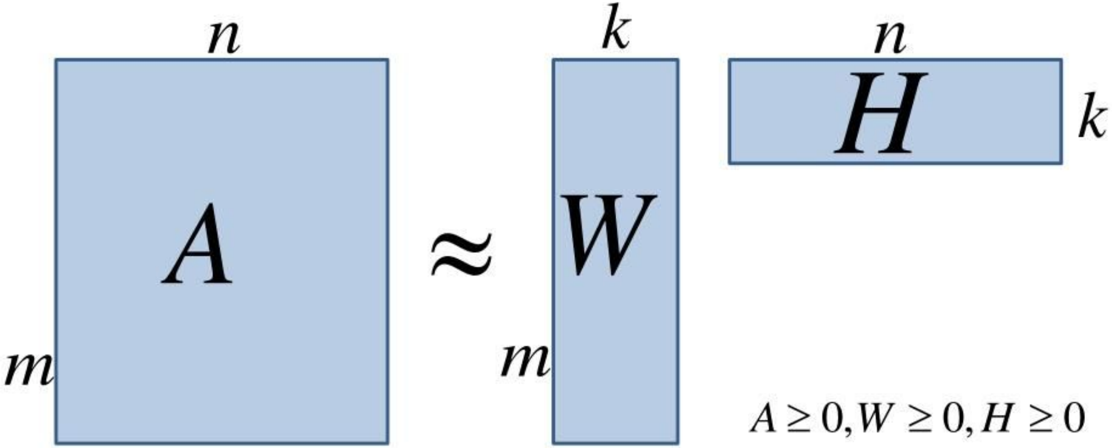
$$\ln p(U,V|R,\sigma^2,\sigma_V^2,\sigma_U^2)$$

$$= -\frac{1}{2\sigma^2}\sum_{i=1}\sum_{j=1} I_{ij}(R_{ij}-U_i^T V_j)^2 - \frac{1}{2\sigma_U^2}\sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M} V_j^T V_j$$

$$-\frac{1}{2}\left(\left(\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\right)\ln\sigma^2 + ND\ln\sigma_U^2 + MD\ln\sigma_V^2\right) + C, \qquad (3)$$

Ref: Mnih, Andriy, and Russ R. Salakhutdinov. "Probabilistic matrix factorization." Advances in neural information processing systems 20 (2007).

Non-Negative Matrix Factorization

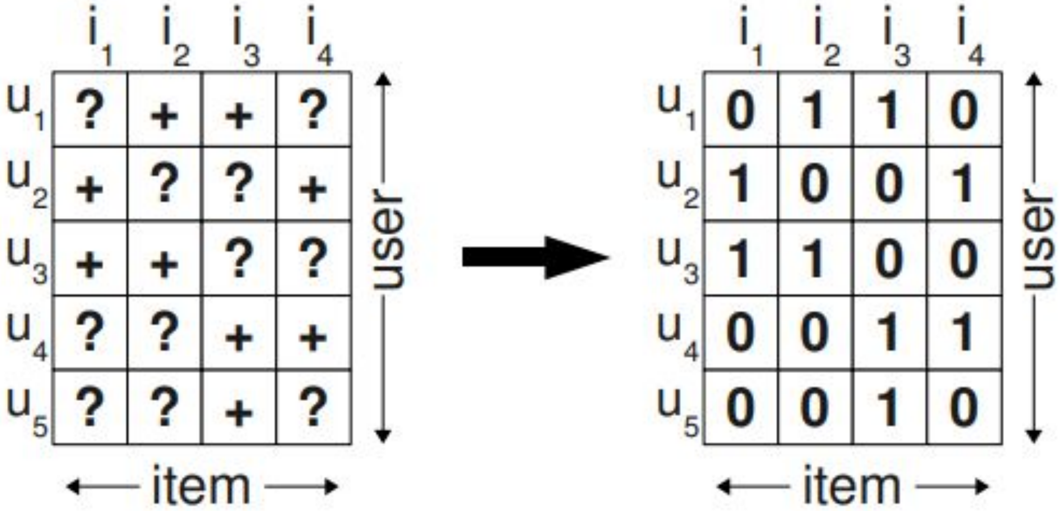


$$H \leftarrow H .* \frac{W^T A}{W^T W H}$$

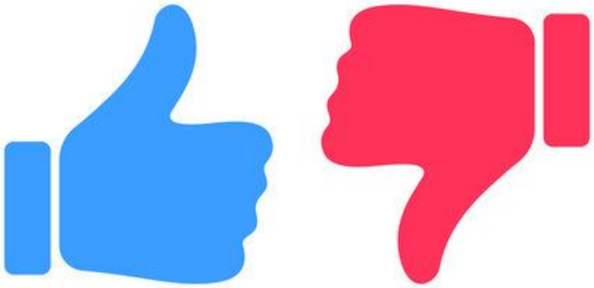
$$W \leftarrow W .* \frac{A H^T}{W H H^T}$$

Ref: Lee, Daniel, and H. Sebastian Seung. "Algorithms for non-negative matrix factorization." Advances in neural information processing systems 13 (2000).

Implicit Association



Implicit Association - Like, Click



Weighted Matrix Factorization

$$\min_{x_\star, y_\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

$$c_{ui} = 1 + \alpha r_{ui}$$

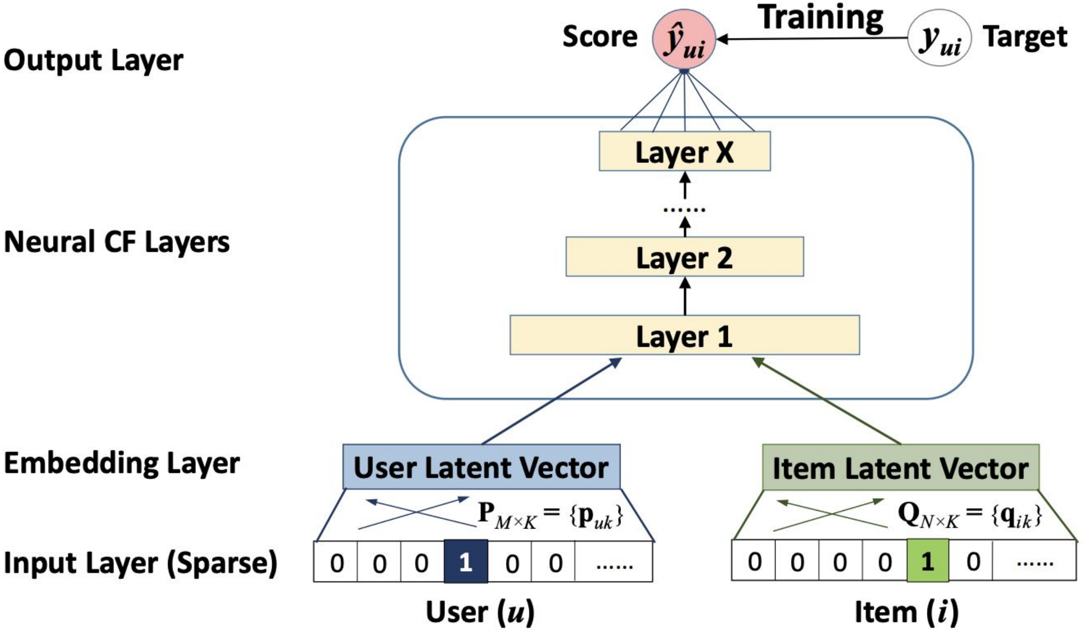
Or

$$\alpha = 40$$

$$c_{ui} = 1 + \alpha \log(1 + r_{ui}/\epsilon)$$

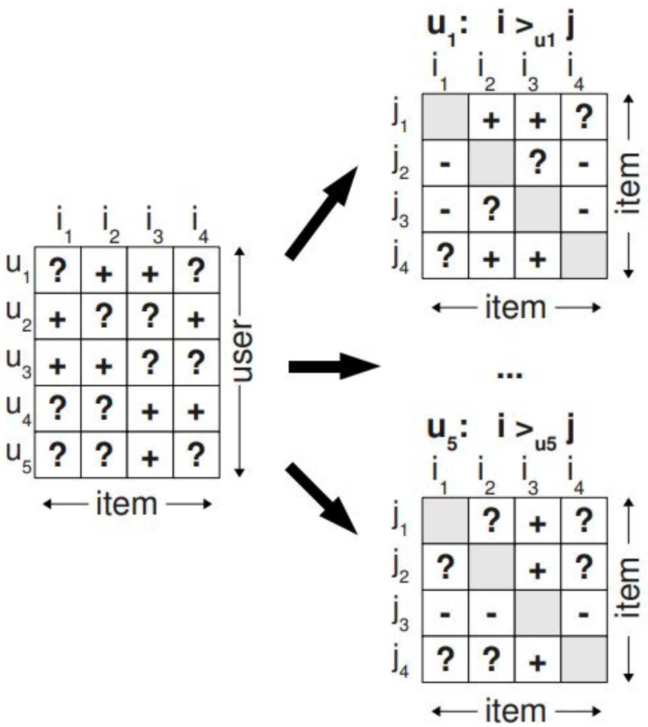
Ref: Hu, Yifan, Yehuda Koren, and Chris Volinsky. "Collaborative filtering for implicit feedback datasets." 2008 Eighth IEEE international conference on data mining. Ieee, 2008.

Neural Collaborative Filtering



Ref: He, Xiangnan, et al. "Neural collaborative filtering." Proceedings of the 26th international conference on world wide web. 2017.

Bayesian Personalized Ranking (BPR) (1)



$$p(\Theta| >_u) \propto p(>_u |\Theta)\, p(\Theta)$$

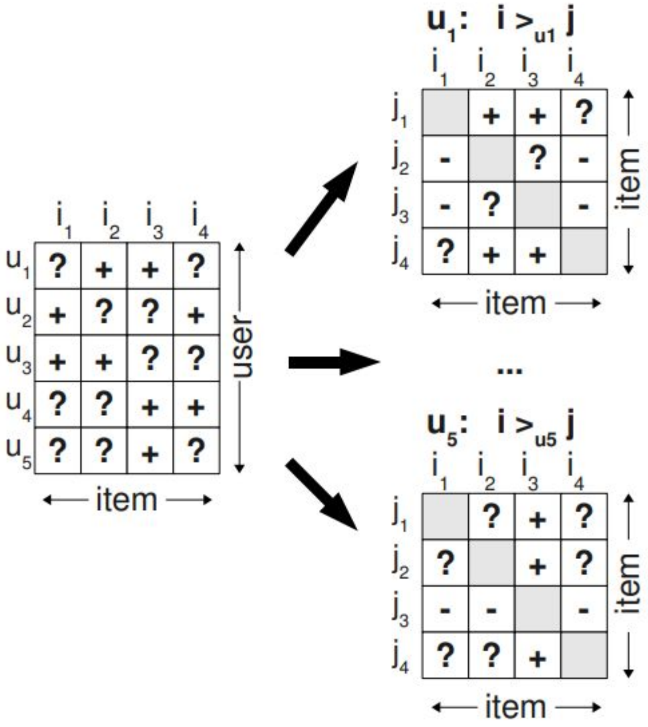
$$\prod_{u \in U} p(>_u |\Theta) = \prod_{(u,i,j) \in D_S} p(i >_u j|\Theta)$$

$$p(\Theta) \sim N(0, \Sigma_\Theta)$$

$$\begin{aligned}\text{BPR-O}\textsc{pt} &:= \ln p(\Theta| >_u) \\ &= \ln p(>_u |\Theta)\, p(\Theta)\end{aligned}$$

Ref: Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback.

Bayesian Personalized Ranking (BPR) (2)



$$p(i >_u j|\Theta) := \sigma(\hat{x}_{uij}(\Theta))$$

$$\hat{x}_{uij} := \hat{x}_{ui} - \hat{x}_{uj}$$

$$\hat{x}_{ui} = \langle w_u, h_i \rangle = \sum_{f=1}^{k} w_{uf} \cdot h_{if}$$

Or

$$\hat{x}_{ui} = \sum_{l \in I_u^+ \wedge l \neq i} c_{il} \qquad c_{i,j}^{\text{cosine}} := \frac{|U_i^+ \cap U_j^+|}{\sqrt{|U_i^+| \cdot |U_j^+|}}$$

Ref: Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. BPR: Bayesian Personalized Ranking from Implicit Feedback.

Weighted Bayesian Personalized Ranking (WBPR)

$$\mathrm{WBPR}(\mathcal{D}_S) = \operatorname*{argmax}_{\Theta} \sum_{(u,i,j)\in\mathcal{D}_S} w_u w_i w_j \ln \sigma(\hat{s}_{u,i,j}) - \lambda \|\Theta\|^2,$$

$$w_j = \sum_{u\in\mathcal{U}} \delta(j \in \mathcal{I}_u^+) \qquad w_u = \frac{1}{|\mathcal{I}_u^+|}$$

Ref: Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. Personalized Ranking for non-uniformly sampled items. KDD Cup 2011

Social Bayesian Personalized Ranking (SBPR)

$$\sum_{u}\left[\sum_{i\in P_u}\sum_{k\in SP_u}\ln(\sigma(\frac{x_{ui}-x_{uk}}{1+s_{uk}}))+\sum_{k\in SP_u}\sum_{j\in N_u}\ln(\sigma(x_{uk}-x_{uj}))\right]-regularization$$

Ref: Zhao, Tong, Julian McAuley, and Irwin King. "Leveraging social connections to improve personalized ranking for collaborative filtering." CIKM. 2014.

Collaborative less-is-more Filtering (CLiMF)

$$RR_i = \sum_{j=1}^{N} \frac{Y_{ij}}{R_{ij}} \prod_{k=1}^{N} (1 - Y_{ik}\mathbb{I}(R_{ik} < R_{ij}))$$

$$\Downarrow$$

$$RR_i \approx \sum_{j=1}^{N} Y_{ij} g(f_{ij}) \prod_{k=1}^{N} (1 - Y_{ik} g(f_{ik} - f_{ij}))$$

$$\frac{1}{R_{ij}} \approx g(f_{ij}) \qquad \mathbb{I}(R_{ik} < R_{ij}) \approx g(f_{ik} - f_{ij})$$

Ref: Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. 2012. CLiMF: learning to maximize reciprocal rank with collaborative less-is-more filtering. RecSys '12

Extended Collaborative less-is-more Filtering (xCLiMF)

$$ERR_m = \sum_{i=1}^{N} r_{mi} g(f_{mi}) \prod_{j=1}^{N} \left(1 - r_{mj} g(f_{m(j-i)})\right)$$

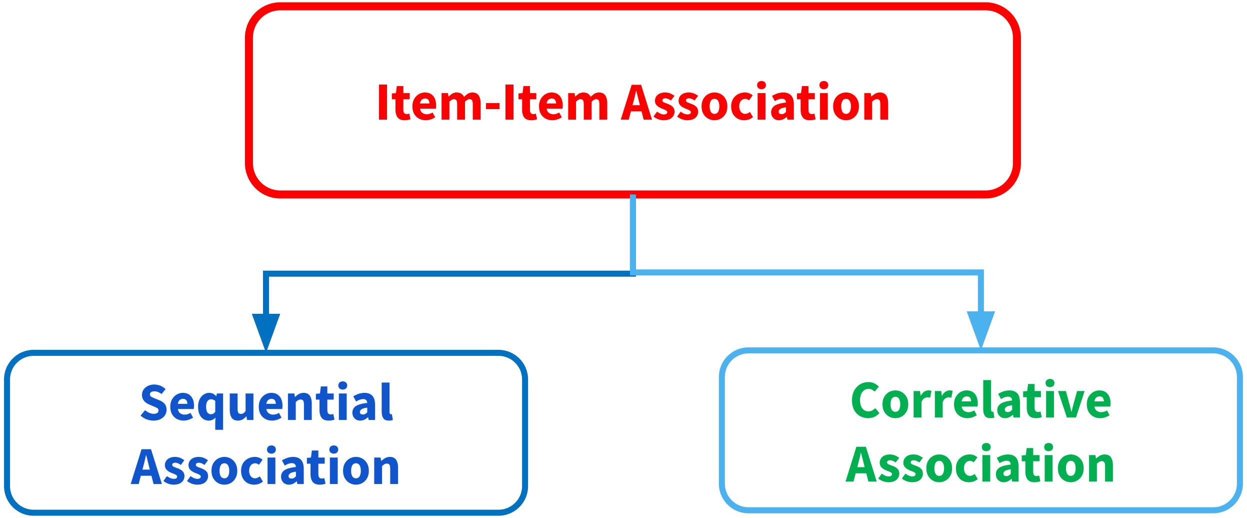
$$r_{mi} = \begin{cases} \frac{2^{y_{mi}} - 1}{2^{y_{max}}}, & I_{mi} > 0 \\ 0, & I_{mi} = 0 \end{cases}$$

Ref: Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, and Alan Hanjalic. 2013. XCLiMF: optimizing expected reciprocal rank for data with multiple levels of relevance. RecSys '13

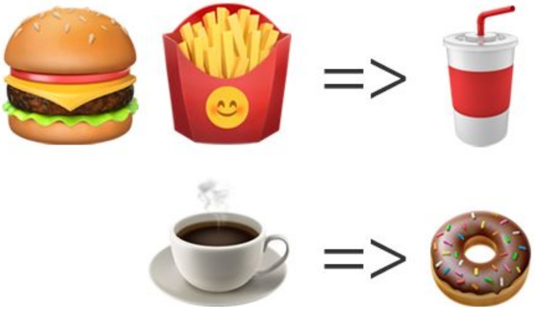
2.

Modeling Item-Item Associations

Recap: Item-Item Association



Correlative Association



Co-occurrence Matrix Factorization

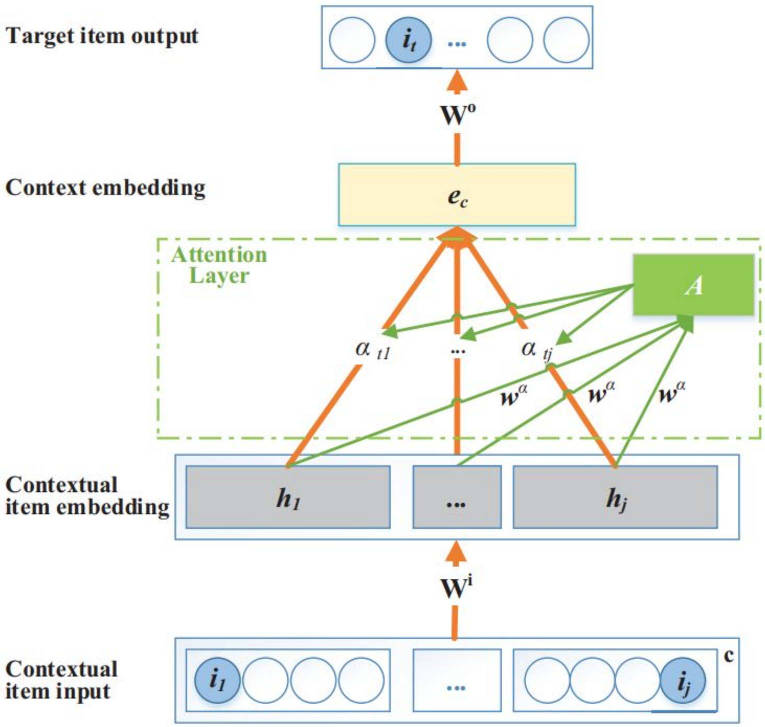
$$\mathcal{L}_{\text{co}} = \overbrace{\sum_{u,i} c_{ui}(y_{ui} - \theta_u^\top \beta_i)^2}^{\text{MF}} + \overbrace{\sum_{m_{ij} \neq 0} (m_{ij} - \beta_i^\top \gamma_j - w_i - c_j)^2}^{\text{item embedding}}$$
$$+ \lambda_\theta \sum_u \|\theta_u\|_2^2 + \lambda_\beta \sum_i \|\beta_i\|_2^2 + \lambda_\gamma \sum_j \|\gamma_j\|_2^2$$

$\theta_u, \beta_i, \gamma_j$ are latent vectors of user u, item i, context j

m_{ij} is pre-computed using point-wise mutual information

Ref: D.Liang et al. Factorization Meets the Item Embedding: Regularizing Matrix Factorization with Item Co-occurrence. RecSys '16

Attention-Based Transactional Context Embedding

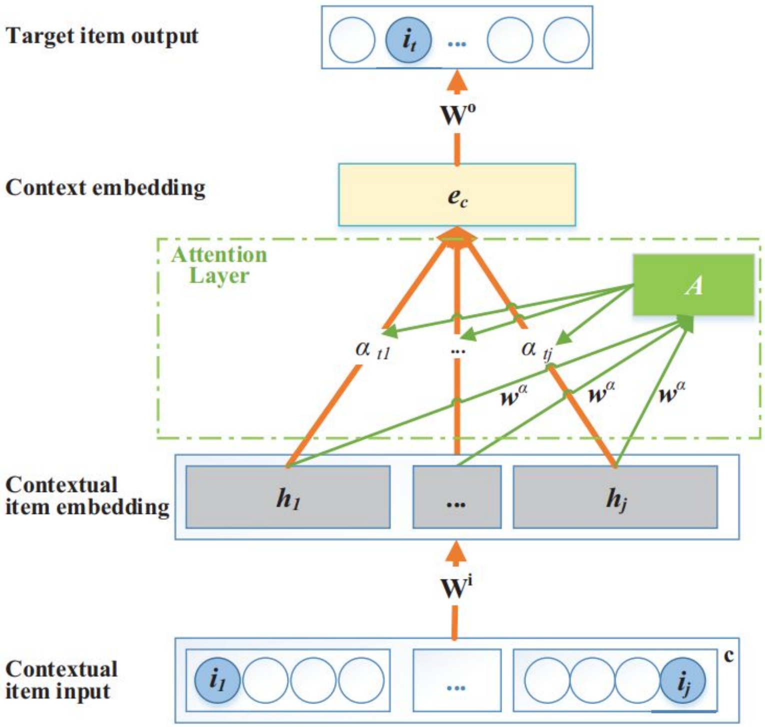


$$\alpha_{tj} = \frac{exp(e(\mathbf{h}_j))}{\sum_{s\in \mathbf{c}_t} exp(e(\mathbf{h}_s))}$$

$$e(\mathbf{h}_j) = \mathbf{w}^\alpha \mathbf{h}_j^T$$

Ref: S. Wang et al. Attention-Based Transactional Context Embedding for Next-Item Recommendation AAAI 2018

Attention-Based Transactional Context Embedding

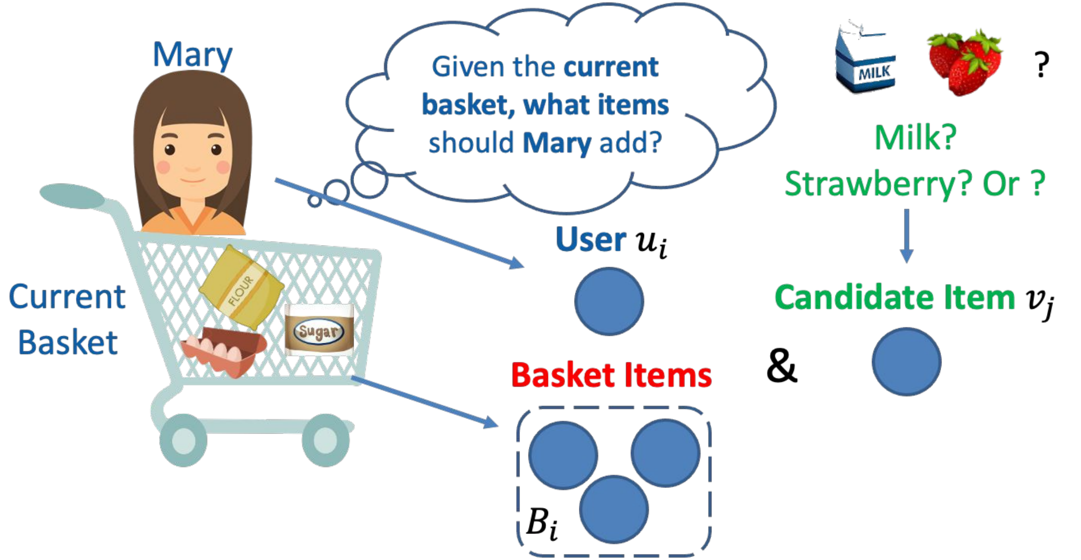


$$\alpha_{tj} = \frac{exp(e(\mathbf{h}_j))}{\sum_{s \in \mathbf{c}_t} exp(e(\mathbf{h}_s))}$$

$$e(\mathbf{h}_j) = \mathbf{w}^\alpha \mathbf{h}_j^T$$

Ref: S. Wang et al. Attention-Based Transactional Context Embedding for Next-Item Recommendation AAAI 2018

Basket-Sensitive Personalized Item Recommendation



Learn a real-value function to rank candidate items

$$F(u_i, B_i, v_j; \Theta)$$

Ref: Duc-Trong Le et al. Basket-Sensitive Personalized Item Recommendation. IJCAI 2017

Basket-Sensitive Personalized Item Recommendation

$$F(u_i, B_i, v_j; \Theta) \propto \boxed{\gamma_1} \cdot x_i^T y_j + \boxed{\gamma_2} \cdot \sum_{v_k \in B_i} y_j^T z_k$$

Adoption Estimation

User & Candidate Item

Candidate Item & Basket Items

$$+ \boxed{\gamma_3} \cdot \sum_{(v_k \neq v_{k'}) \in B_i} z_k^T z_{k'} + \boxed{\gamma_4} \cdot \sum_{v_k \in B_i} x_i^T z_k$$

Among Basket Items

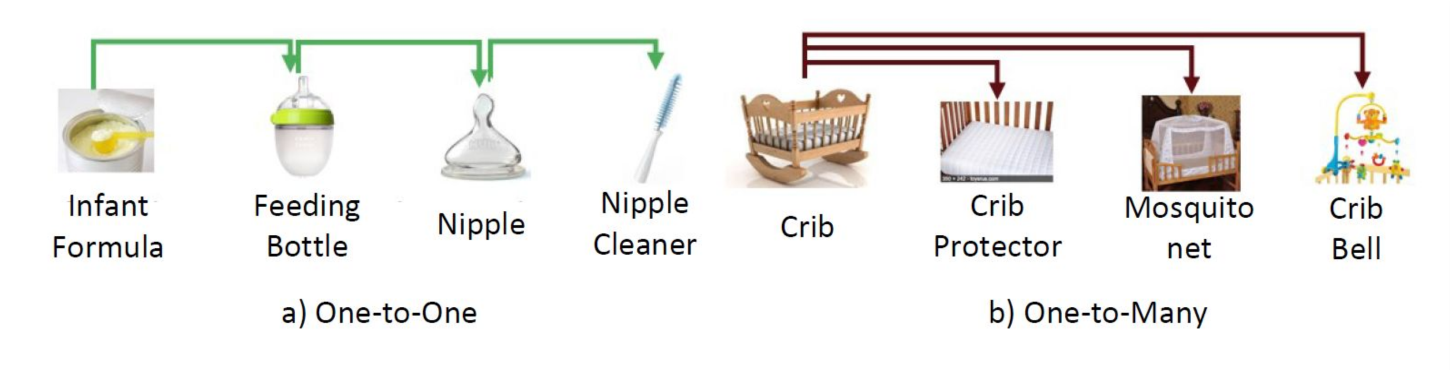
User & Basket Items

where $x_i, y_j, z_k \in \mathbb{R}^K$ are latent vectors; $\gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \{0,1\}$

Ref: Duc-Trong Le et al. Basket-Sensitive Personalized Item Recommendation. IJCAI 2017

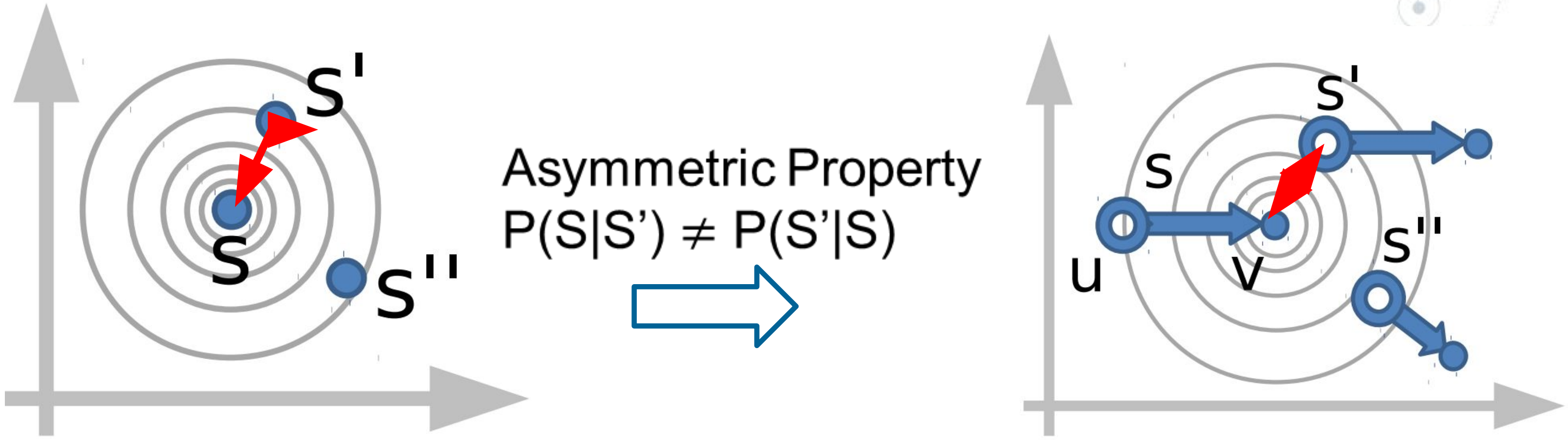
Sequential Association

Data: Sequence – Items are adopted sequentially by time.



Hypothesis: The next selection (item) of a user is affected by his preceding adoptions

Metric Embedding



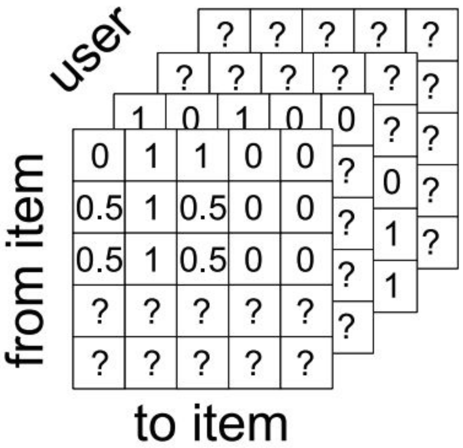
Single-Point Model

Dual-Point Model

$$\Pr(p^{[i]}|p^{[i-1]}) = \frac{e^{-||X(p^{[i]})-X(p^{[i-1]})||_2^2}}{\sum_{j=1}^{|S|} e^{-||X(s_j)-X(p^{[i-1]})||_2^2}}$$

Ref: S. Chen et al. Playlist Prediction via metric embedding. KDD 2012

Factorizing Personalized Markov Chains



$$p(i \in B_t^u | B_{t-1}^u) = \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} p(i \in B_t^u | l \in B_{t-1}^u)$$

$$\hat{p}(i \in B_t^u | B_{t-1}^u) = \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} \hat{a}_{u,l,i}$$

$$= \frac{1}{|B_{t-1}^u|} \sum_{l \in B_{t-1}^u} (\langle v_u^{U,I}, v_i^{I,U} \rangle + \langle v_i^{I,L}, v_l^{L,I} \rangle$$

$$+ \langle v_u^{U,L}, v_l^{L,U} \rangle)$$

Ref: S. Rendle et al. Factorizing personalized Markov chains for next-basket recommendation. WWW 2010

Factorization Machines

Feature vector **x** / Target y

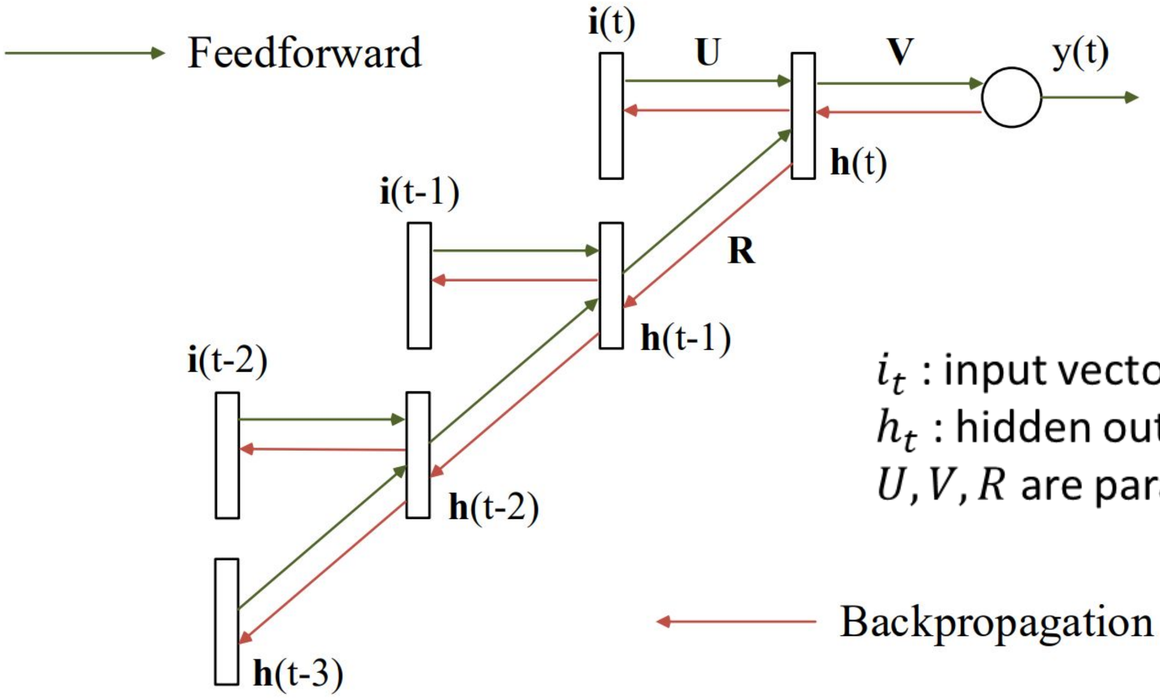
	User				Movie					Other Movies rated					Time	Last Movie rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...	Time	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...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

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

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

Ref: S. Render et al. Factorization Machines. KDD 2012

Recurrent Neural Networks



Click probability

i_t : input vector at time t
h_t : hidden output at time t
U, V, R are parameters

Ref: Y. Zhang et al. Sequential Click Prediction for Sponsored Search with RNNs. AAAI 2014

The Notion of Basket Sequence

Session t0

Basket

Session t1

Basket

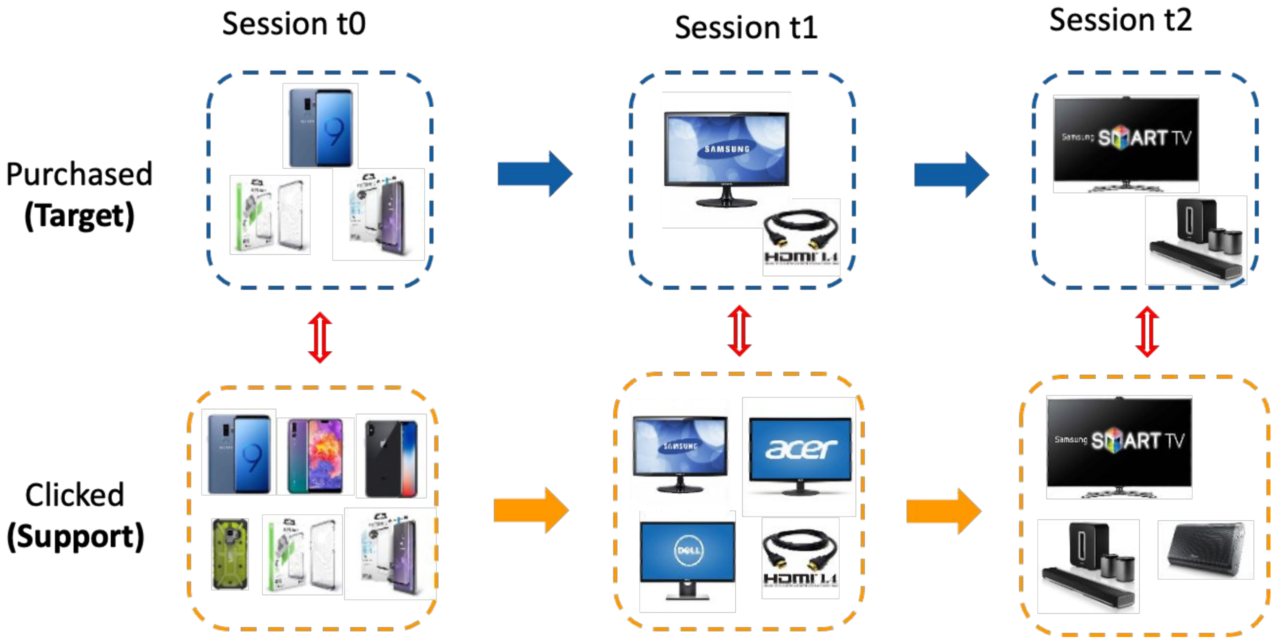
Session t2

Basket

Correlative associations among items in a basket

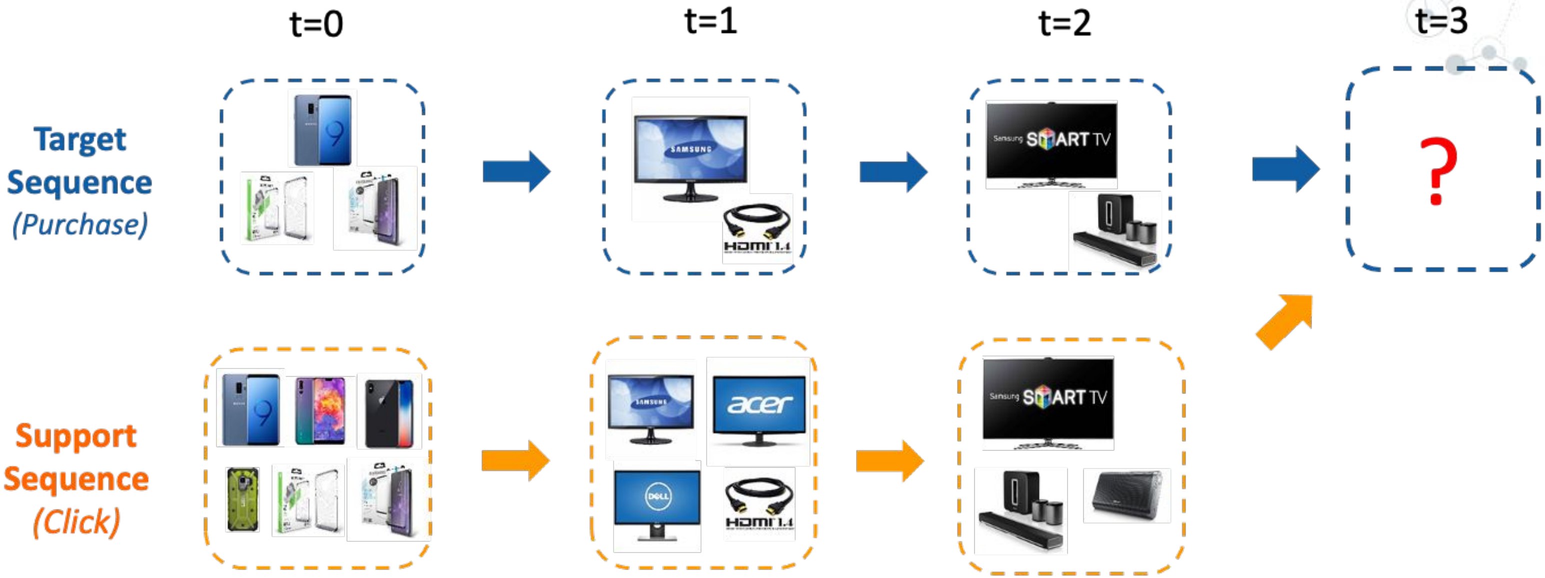
Sequential associations across baskets in a sequence

Contemporaneous Basket Sequences (CBS)



Ref: Duc-Trong Le et al. Modeling contemporaneous basket sequences with twin networks for next-item recommendation. IJCAI 2018

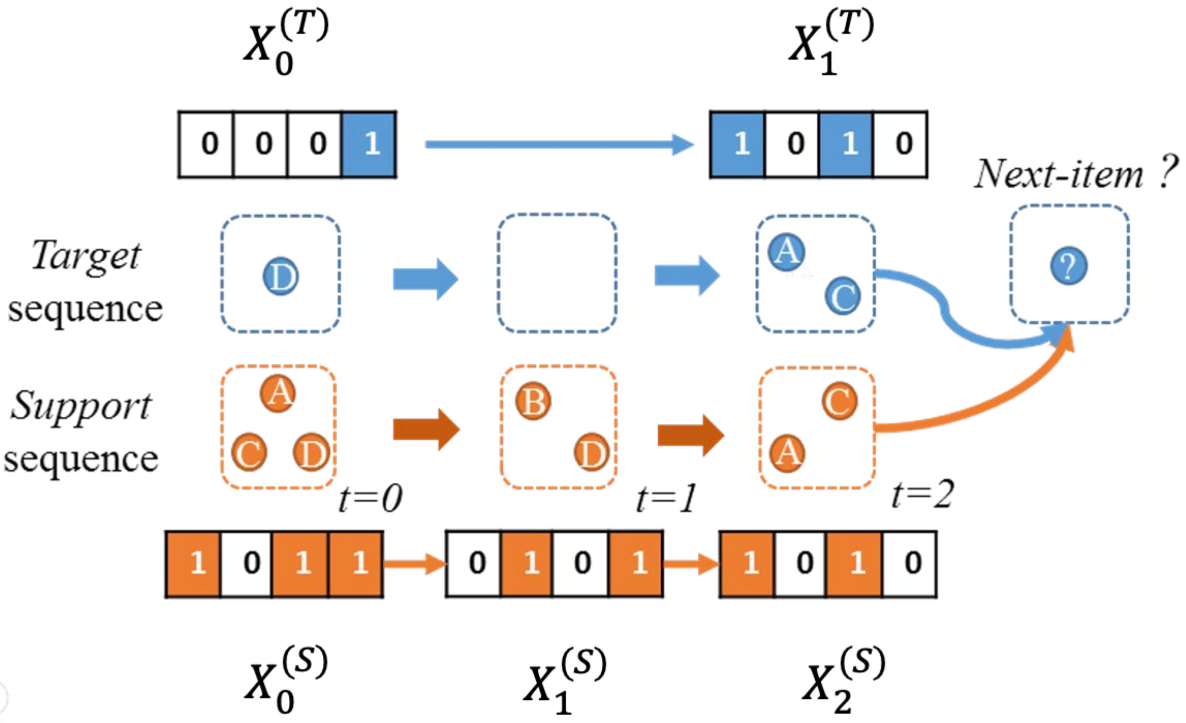
Next-Item Recommendation with CBS



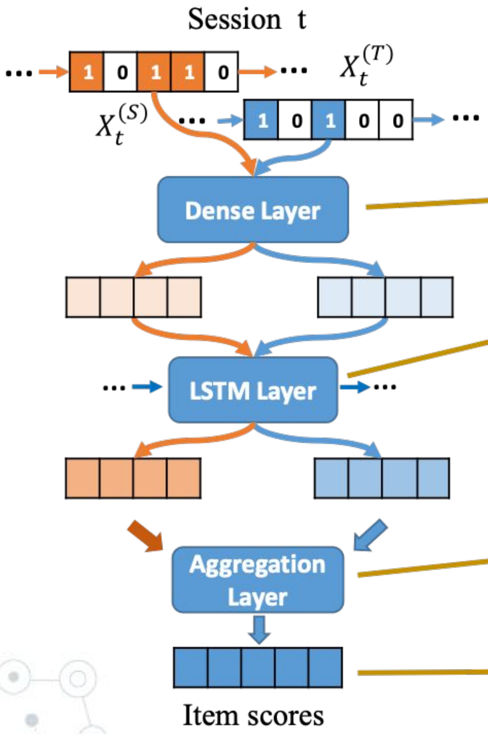
TASK:

Modeling **correlative** & **sequential** associations in **CBS** concurrently to predict the **next "target" item**

Next-Item Recommendation with CBS



CBS with Siamese Networks (CBS-SN)



- **Hypothesis**: CBS reflect the same underlying behavior
- **Formulation:** $X_t \in \{0,1\}^N$

$$b_t = f(\Theta_b X_t + \Omega_b)$$

$$\Theta_b \in \mathbb{R}^{L \times N}, \Omega_b \in \mathbb{R}^L$$

$$h_t = g(\Phi_b b_t + \Phi_h h_{t-1} + \Omega_h)$$

$$\Phi_b \in \mathbb{R}^{H \times L}, \Phi_h \in \mathbb{R}^{H \times H}, \Omega_h \in \mathbb{R}^H$$

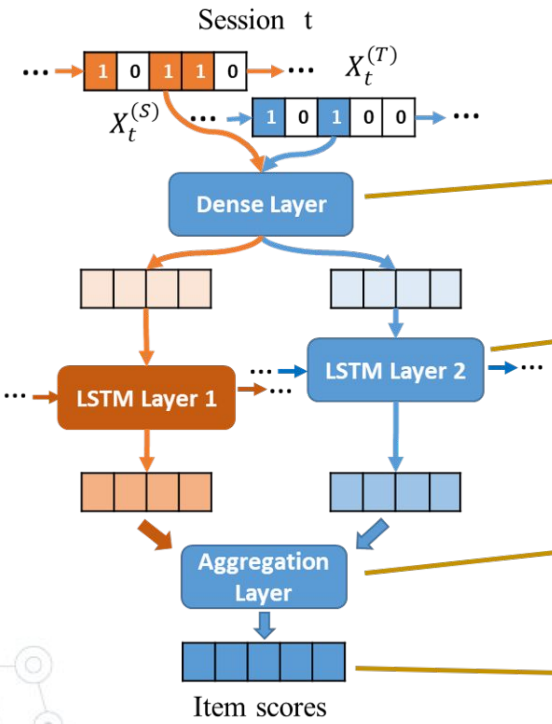
$$\hat{h} = concat(\hat{h}_T, \hat{h}_S)$$

$$r_{\text{agg}} = W.\hat{h} + \Omega_c$$

$$W \in \mathbb{R}^{N \times 2H}, \Omega_c \in \mathbb{R}^N$$

$$Y = softmax(r_{\text{agg}})$$

CBS with Concordant Fraternal Networks (CBS-SN)



- **Hypothesis**: CBS reflect different long-term sequential dependencies
- **Formulation**: $X_t \in \{0,1\}^N$

$$b_t = f(\Theta_b X_t + \Omega_b)$$

$$\Theta_b \in \mathbb{R}^{L\times N}, \Omega_b \in \mathbb{R}^L$$

$$h_t^{(S)} = g(\Phi_b^{(S)} b_t^{(S)} + \Phi_h^{(S)} h_{t-1}^{(S)} + \Omega_h^{(S)})$$

$$h_t^{(T)} = g(\Phi_b^{(T)} b_t^{(T)} + \Phi_h^{(T)} h_{t-1}^{(T)} + \Omega_h^{(T)})$$

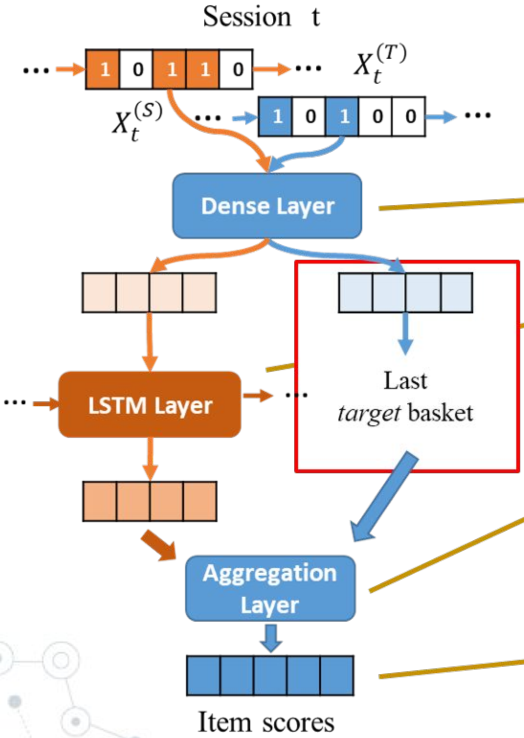
$$\Phi_b^{(*)} \in \mathbb{R}^{H\times L}, \Phi_h^{(*)} \in \mathbb{R}^{H\times H}, \Omega_h^{(*)} \in \mathbb{R}^H$$

$$r_{agg} = W_1.\hat{h}_S + W_2.\hat{h}_T + \Omega_c$$

$$W_1, W_2 \in \mathbb{R}^{N\times H}, \Omega_c \in \mathbb{R}^N$$

$$Y = softmax(r_{agg})$$

CBS with Discordant Fraternal Networks (CBS-SN)



- **Hypothesis**: Short-term dependencies in "target" while long-term in "support"
- **Formulation**: $X_t \in \{0,1\}^N$

$$b_t = f(\Theta_b X_t + \Omega_b)$$

$$\Theta_b \in \mathbb{R}^{L \times N}, \Omega_b \in \mathbb{R}^L$$

$$h_t = g(\Phi_b b_t + \Phi_h h_{t-1} + \Omega_h)$$

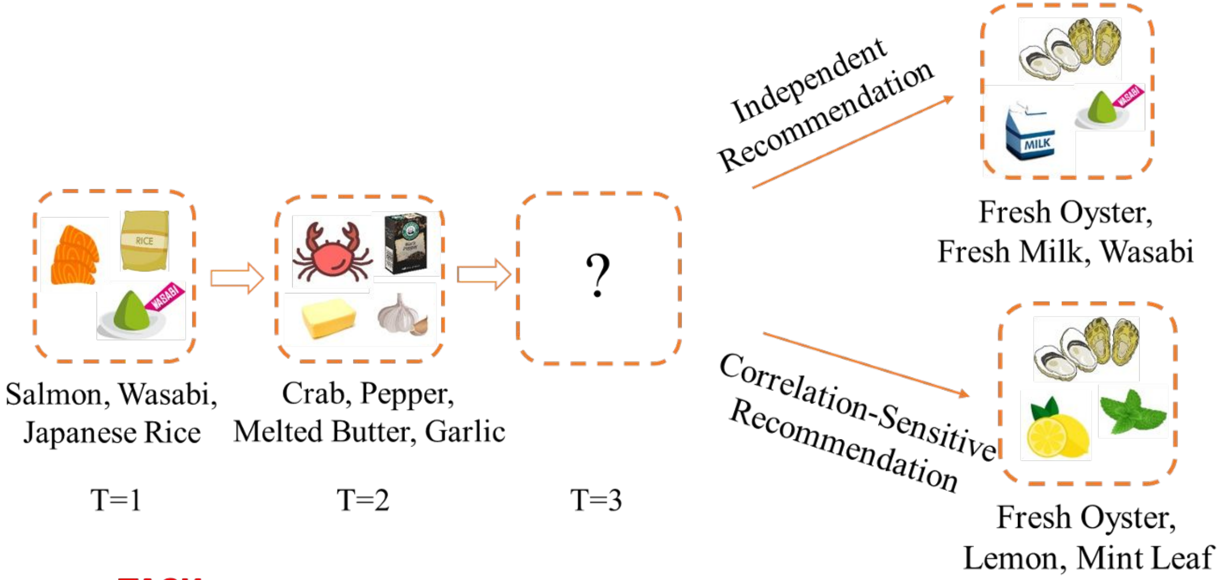
$$\Phi_b \in \mathbb{R}^{H \times L}, \Phi_h \in \mathbb{R}^{H \times H}, \Omega_h \in \mathbb{R}^H$$

$$\boxed{r_{\text{agg}} = W_1.\hat{h}_S + W_2 \hat{b}_T + \Omega_c}$$

$$W_1 \in \mathbb{R}^{N \times H}, W_2 \in \mathbb{R}^{N \times L}, \Omega_c \in \mathbb{R}^N$$

$$Y = \mathit{softmax}(r_{\text{agg}})$$

Correlation-Sensitive Next-Basket Recommendation

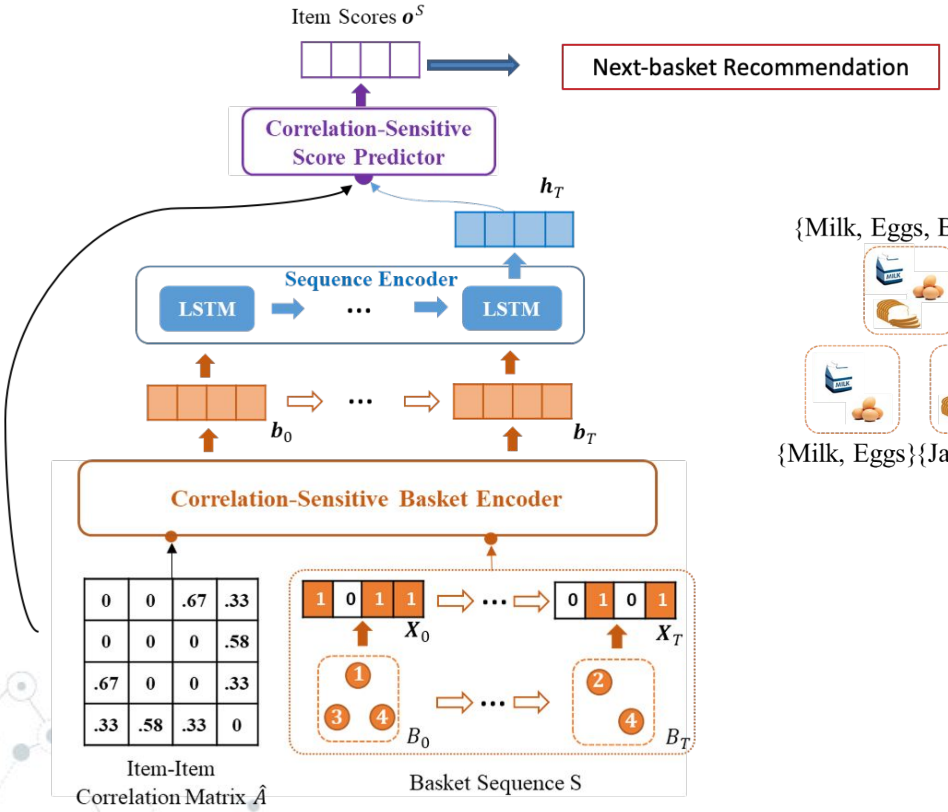


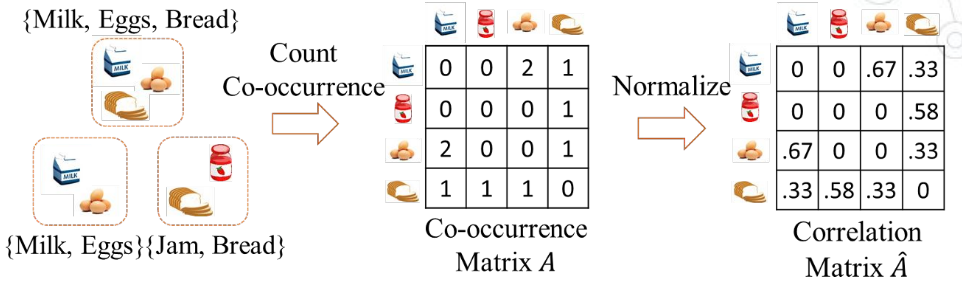
TASK:

Modeling concurrently **correlative** & **sequential** associations in basket sequences to predict **the next-basket of correlated items.**

Ref: Duc-Trong Le et al. Correlation-Sensitive Next-Basket Recommendation. IJCAI 2019

Basket-Sequence Correlation Networks (Beacon)

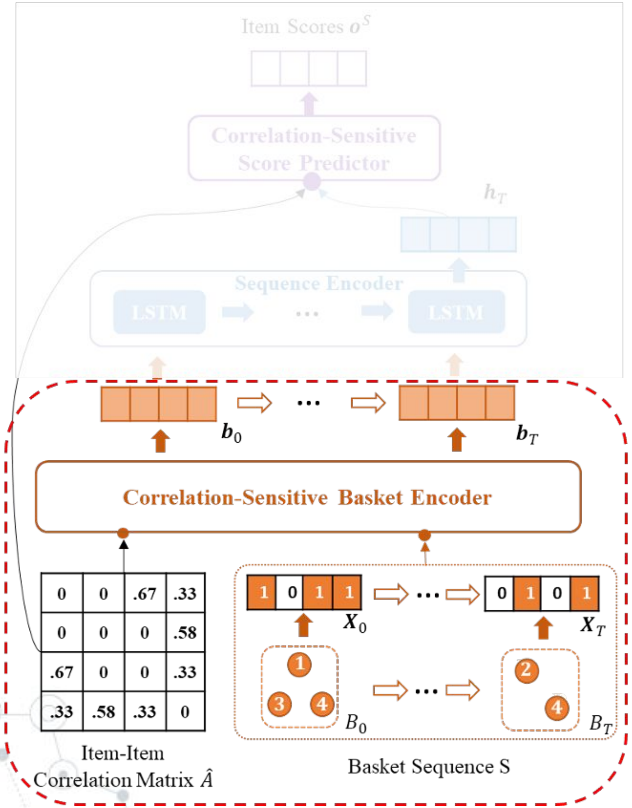




$$\hat{A} = \tilde{D}^{-\frac{1}{2}} A \tilde{D}^{-\frac{1}{2}}$$

$$\tilde{D}_{ii} = \sum_j A_{ij}.$$

Correlation-Sensitive Basket Encoder



- Input: Given a basket B

$$X \in \{0,1\}^N \qquad \hat{A} \in \mathbb{R}^{N \times N}$$

- The **immediate representation** of B

$$z_t = X \odot \underbrace{\Omega_V}_{\text{Item Importance Parameters}} + ReLU(X\hat{A} - \underbrace{\eta_A}_{\text{Noise-Canceling Parameter}} \mathbf{1})$$

Item Importance Parameters

Noise-Canceling Parameter

$$z_t, \Omega_V \in \mathbb{R}^N \qquad \eta_A \in \mathbb{R}$$

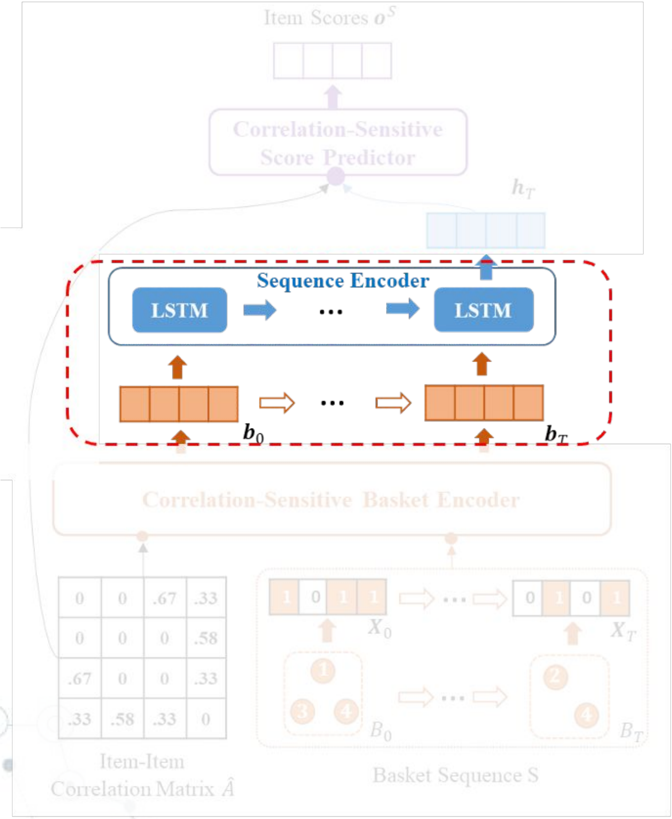
- The **D-dimensional latent representation**:

$$b_t = ReLU(\Phi_z z_t + \Omega_z)$$

$$\Phi_z \in \mathbb{R}^{D \times N}, \Omega_z \in \mathbb{R}^D$$

$$b_t \in \mathbb{R}^D$$

Sequence Encoder



- Input: Given the D-dimensional latent representation of basket B_t at time t

$$b_t \in \mathbb{R}^D$$

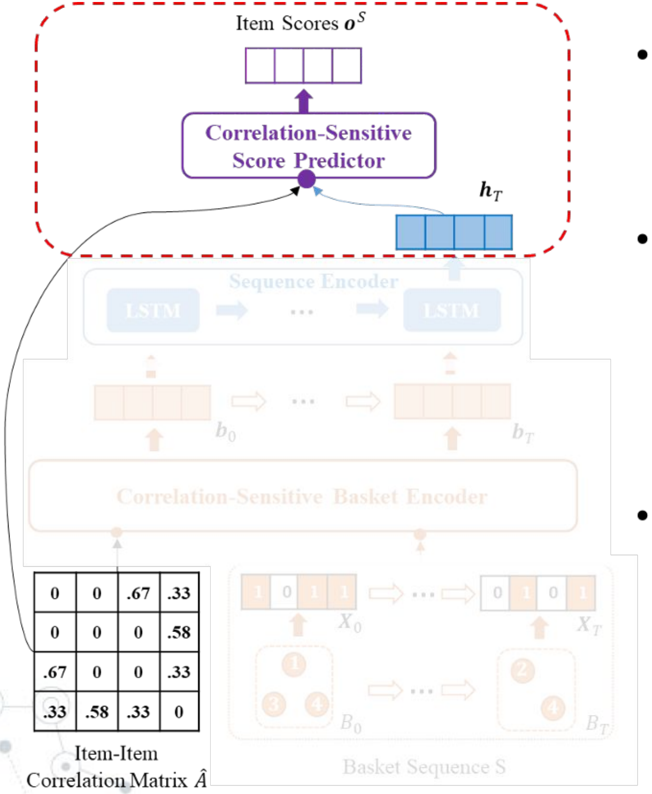
- The **recurrent hidden output** h_t at time t:

$$\boxed{h_t = tanh(\Phi_b b_t + \Phi_h h_{t-1} + \Omega_h)}$$

$$\Phi_b \in \mathbb{R}^{H \times D}, \Phi_h \in \mathbb{R}^{H \times H}, \Omega_h \in \mathbb{R}^H$$

$$h_t \in \mathbb{R}^H$$

Correlation-Sensitive Score Predictor



- Input: Given the hidden output of the last basket B_T:

$$h_T \in \mathbb{R}^H$$

- The **sequential signal** for next-item adoptions:

$$p_V = \sigma(\Phi_S h_T)$$

$$\Phi_S \in \mathbb{R}^{N \times H} \quad p_V \in \mathbb{R}^N$$

- The **correlation-sensitive score** predictor:

$$o^S = \underbrace{\alpha(p_V \odot \Omega_V + p_V.\hat{A})}_{\text{Basket-Sensitive}} + \underbrace{(1-\alpha)p_V}_{\text{Sequence-Sensitive}}$$

$$\alpha \in [0,1] \quad o^S \in \mathbb{R}^N$$

Recommendation & Optimization

- **Correlation-Sensitive Next-Basket Recommendation:**
  - The predicted ranking $\hat{r}^S$ of the item set based on o^S

    $$\hat{r}^S : o^S \to \{1, 2, \dots, N\}$$

    where $\hat{r}_i^S$ is the ranking of item i.
  - Approximately recommended basket B_{T+1} of size K:

    $$B_{T+1}|K = \{i | \hat{r}_i^S \leq K\}$$

- **Optimization:**

$$\boxed{\theta^* = \operatorname{argmin}_{\theta \in \Theta} \sum_{S \in \mathcal{S}} \mathcal{L}_S}$$

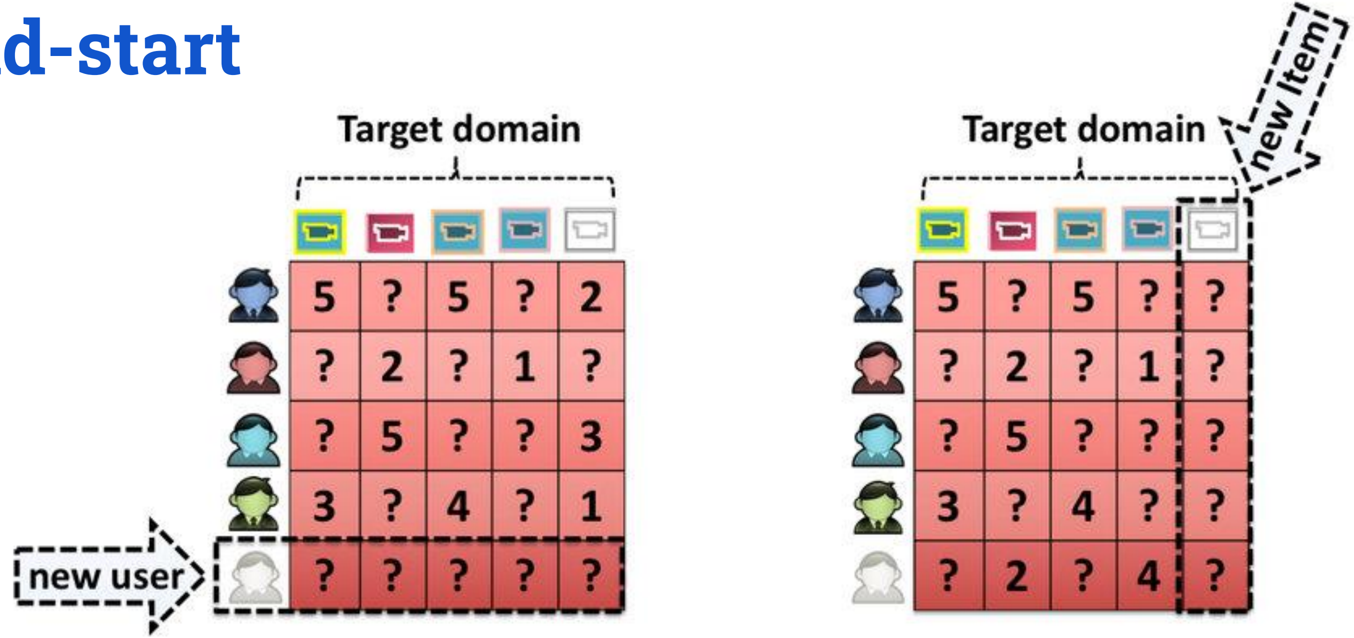
$$\boxed{\mathcal{L}_S = -\sum_{i \in B_T} \ln \sigma(o_i^{S'}) - \frac{|B_T|}{|V'|} \sum_{j \in V'} \ln(1 - \sigma(o_j^{S'} - o_m^{S'}))}$$

$$\boxed{S' = S \setminus B_T, V' = V \setminus \{i \in B_T\}} \quad \boxed{o_m^{S'} = \min\{o_i^{S'} | i \in B_T\}}$$

3.

Challenges & Suggestions

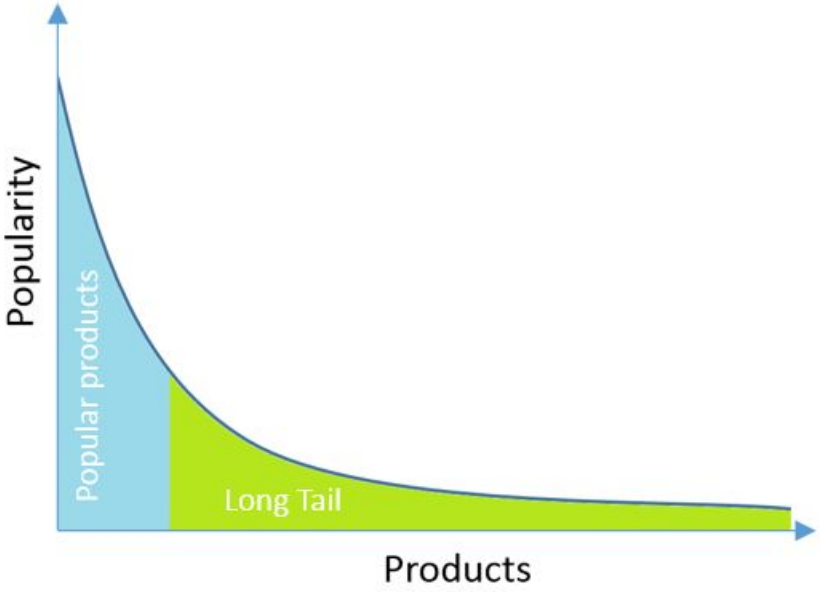
Cold-start



Suggestions:

- Ask users to provide some information (e.g., category selection)
- Initialize user/item vectors with a “mean” vector
- Exploit information from other data sources

Long-tail



Suggestions:

- Content-based strategies
- Weighting constraints
- Correlation

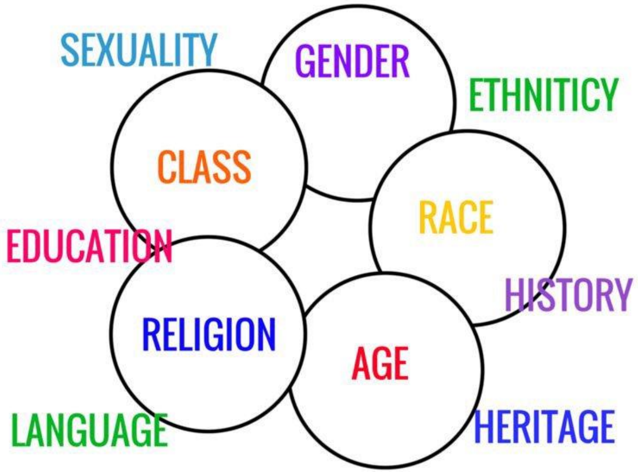
Privacy

Suggestions:

- Anonymization
- Adding noise
- Privacy-Preserving Models

Fairness

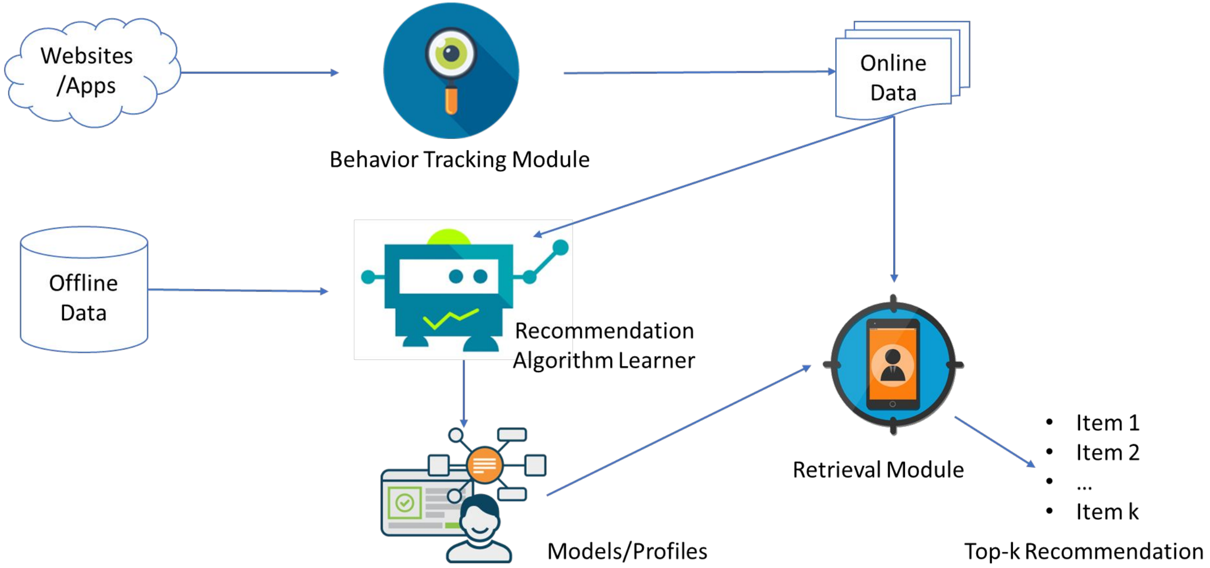
Recommender Systems



Suggestions:

- Remove sensitive attributes
- Fairness Difference Constraint

Online Learning



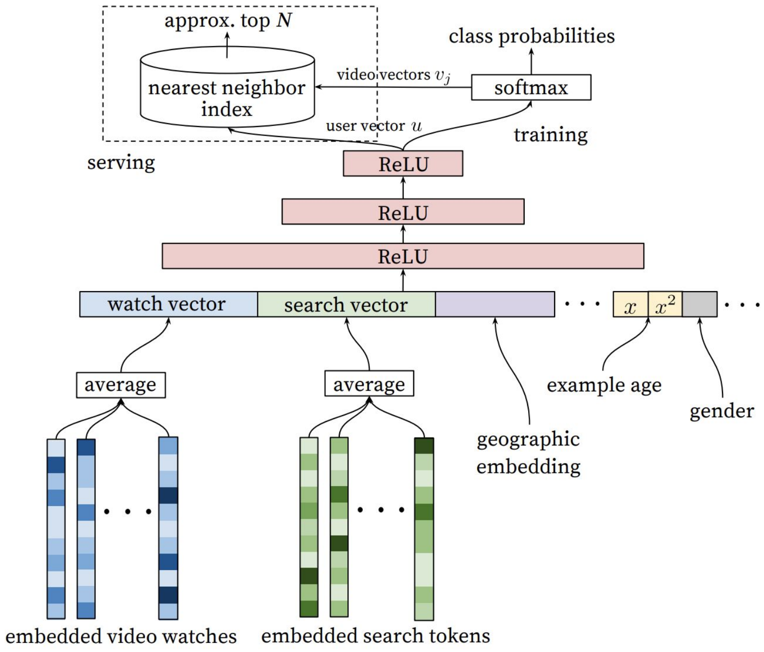
Suggestions:

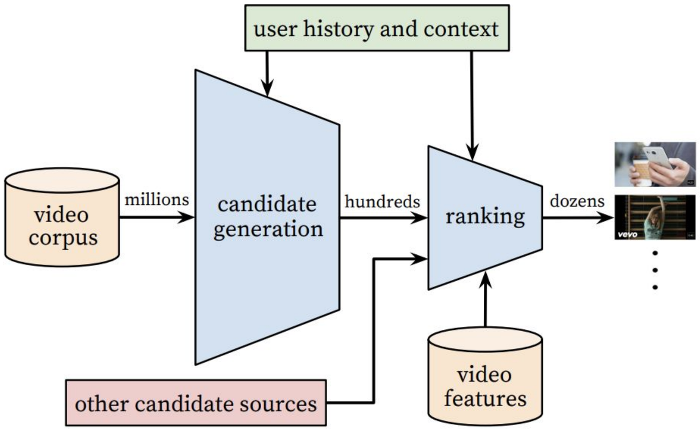
- Non-parameter model
- Iterative learning

4.

Real-life Recommendation Systems

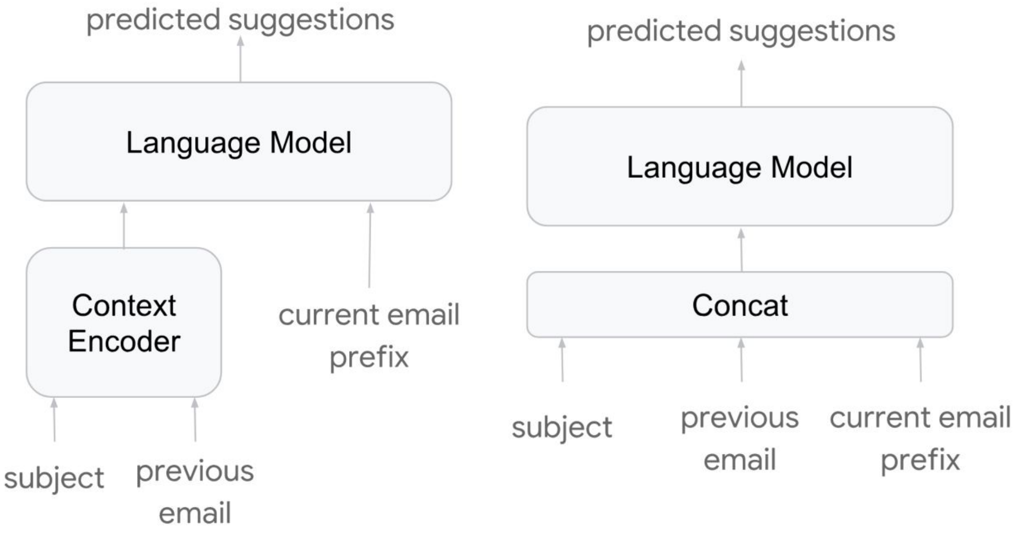
Youtube Recommendation





Ref: Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." RecSys 2016.

Gmail - Real-time Assisted Writing



(a) LM-A (b) LM-B

Ref: Chen, Mia Xu, et al. "Gmail smart compose: Real-time assisted writing." KDD 2019.

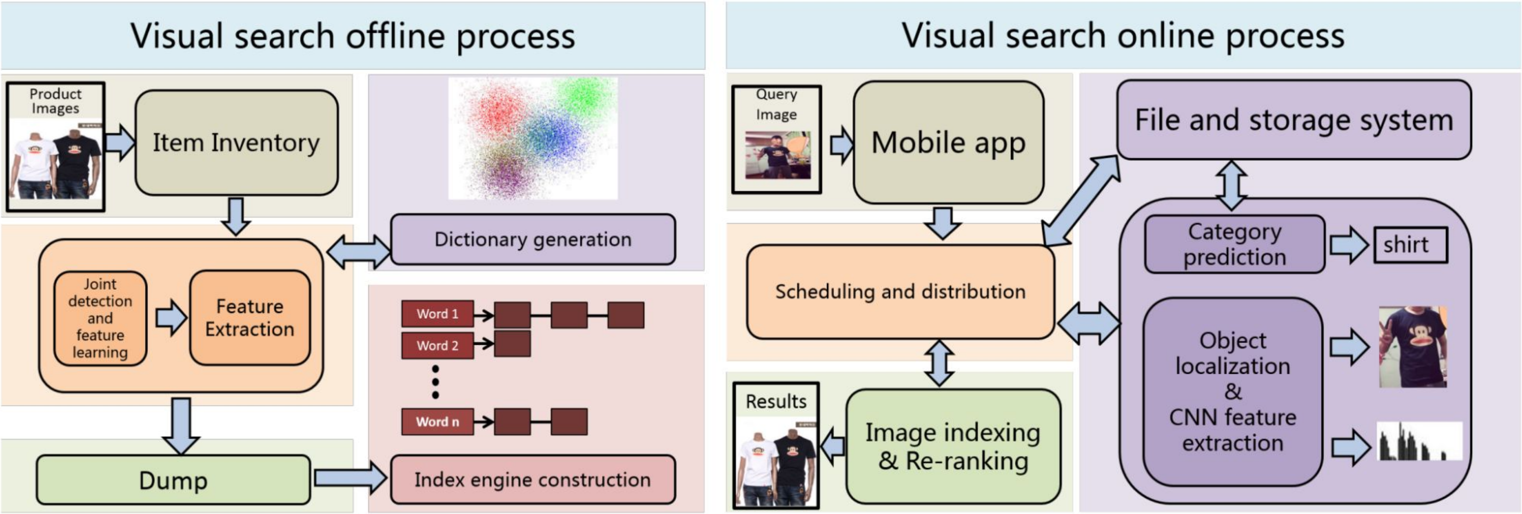
Amazon Recommendation

$$
\begin{aligned}
E_{XY} &= \sum_{c\in X}\left[1-(1-P_Y)^{|c|}\right] = \sum_{c\in X}\left[1-\sum_{k=0}^{|c|}\binom{|c|}{k}(-P_Y)^k\right] \\
&= \sum_{c\in X}\left[1-\left[1+\sum_{k=1}^{|c|}\binom{|c|}{k}(-P_Y)^k\right]\right] = \sum_{c\in X}\sum_{k=1}^{|c|}(-1)^{k+1}\binom{|c|}{k}P_Y^k \\
&= \sum_{c\in X}\sum_{k=1}^{\infty}(-1)^{k+1}\binom{|c|}{k}P_Y^k \qquad \text{(since } \binom{|c|}{k}=0 \text{ for } k>|c|) \\
&= \sum_{k=1}^{\infty}\sum_{c\in X}(-1)^{k+1}\binom{|c|}{k}P_Y^k \qquad \text{(Fubini's theorem)} \\
&= \sum_{k=1}^{\infty}\alpha_k(X)P_Y^k \qquad \text{where } \alpha_k(X)=\sum_{c\in X}(-1)^{k+1}\binom{|c|}{k}.
\end{aligned}
$$

Figure 1. The derivation of the expected number of customers who bought both items X and Y, accounting for multiple opportunities for each X-buyer to buy Y.

Ref: Smith, Brent, and Greg Linden. "Two decades of recommender systems at Amazon. com." IEEE internet computing 21.3 (2017)

Visual Search@Alibaba



Ref: Zhang, Yanhao, et al. "Visual search at alibaba." KDD. 2018.

Tiktok Streaming Engine

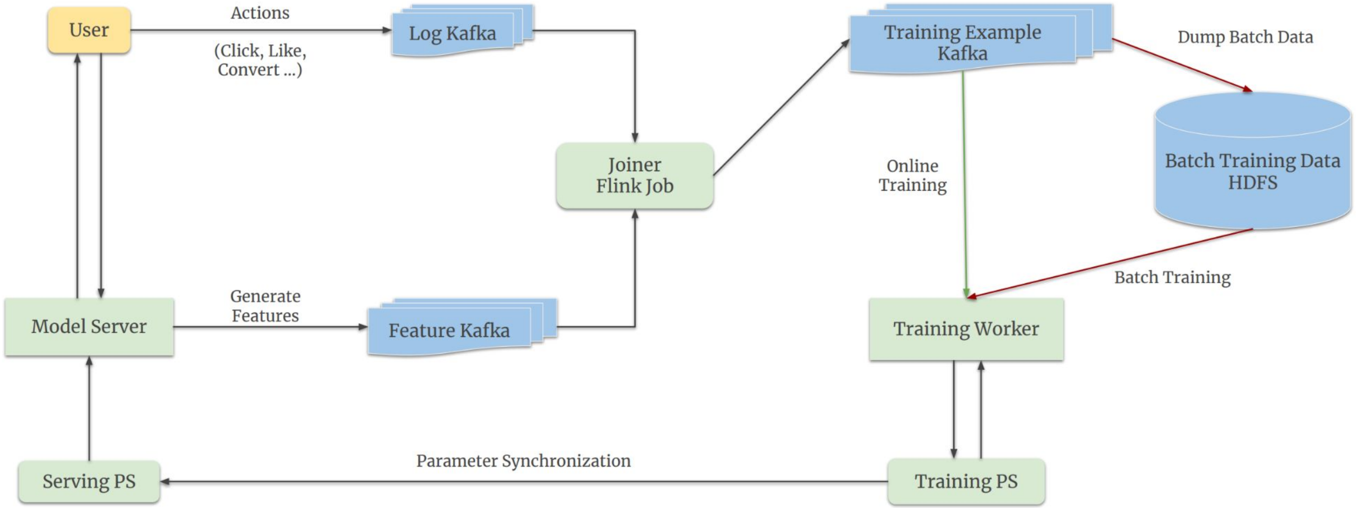


Figure 4: Streaming Engine.
e information feedback loop from [User → Model Server → Training Worker → Model Server → User] would spend a long time when taking the Batch Training path, while the Online Training will close the loop more instantly.

Ref: Liu, Zhuoran, et al. "Monolith: Real Time Recommendation System With Collisionless Embedding Table." Recsys 2022..

Summary

- Modeling User-Item Associations
  - Matrix Factorization & Variants
  - Bayesian Personalized Ranking & Variants
- Modeling Item-Item Associations
  - Latent Factor-based Methods
  - Neural Network-based Methods
- Challenges and Suggestions
- Real-life Recommendation Systems

Thanks!

trongld@vnu.edu.vn