Enhancing Collaborative Filtering with Generative Augmentation

KDD'19

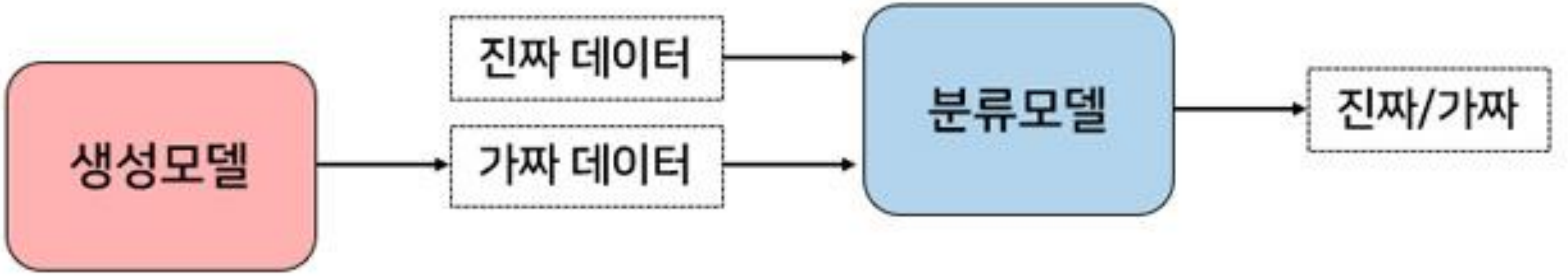
서울시립대학교
UNIVERSITY OF SEOUL

cheon.research@gmail.com

2020. 03. 31.

GAN

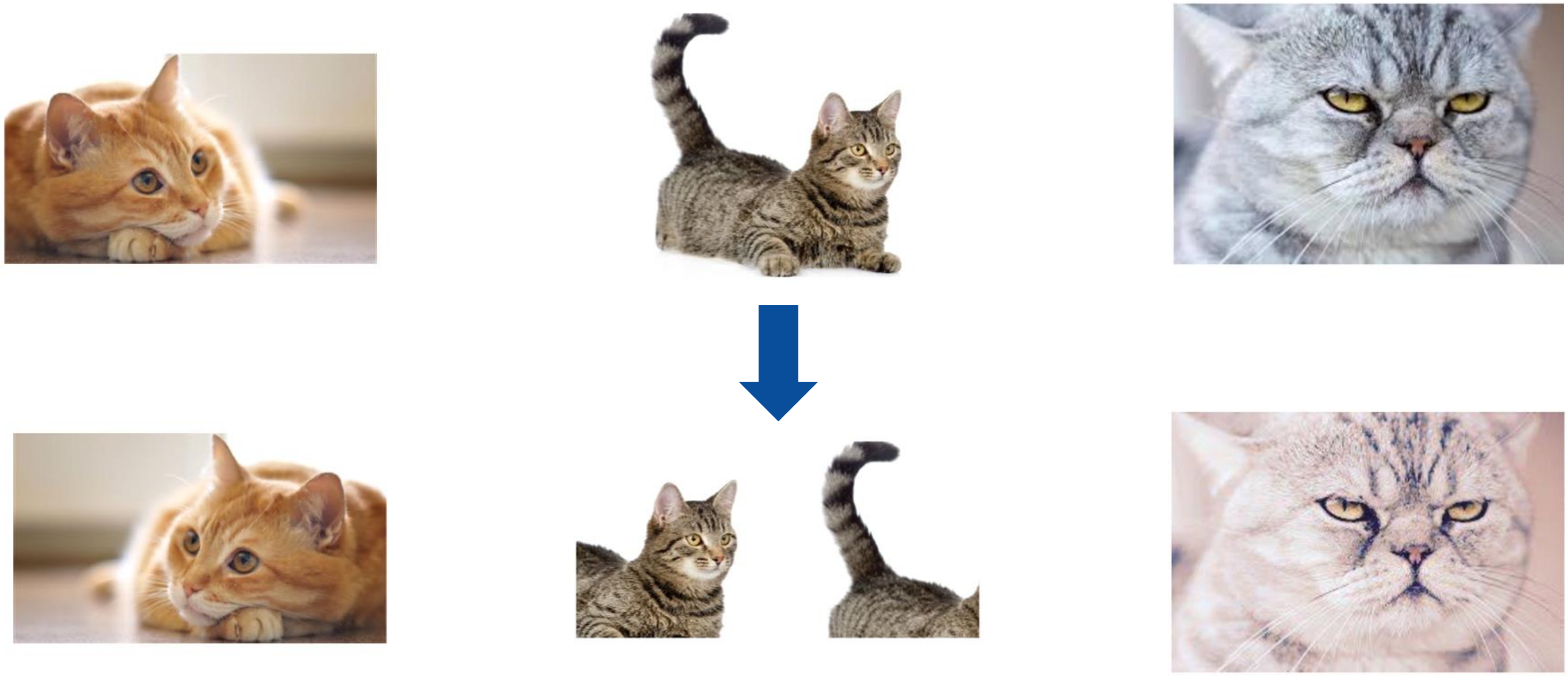
❑ Generative Adversarial Network



$$\min_G \max_D V(D,G) = E_{x \sim p_{data}(x)}[log\ D(x)] + E_{z \sim p_z(z)}[log(1 - D(G(z))]$$

https://www.samsungsds.com/global/ko/support/insights/Generative-adversarial-network-AI-2.html

Data Augmentation



서울시립대학교
UNIVERSITY OF SEOUL

Preliminaries

First, we introduce some important notations used throughout the paper. Let $\mathcal{U}$ and $\mathcal{V}$ be the sets of all users and items, and let M and N be their sizes, respectively. We define the interaction between user u and item v in the dataset C as:

$$c = \begin{cases} 1, & u \text{ likes } v; \\ 0, & u \text{ dislikes } v; \\ \emptyset, & \text{unvisited} \end{cases} \tag{1}$$

Preliminaries

we consider pure CF as a prediction task. Given a user $u \in \mathcal{U}$ and an item $v \in \mathcal{V}$, the score for u likes v is estimated by $f(u, v, c; \mathbf{P}, \mathbf{Q})$, where f is a non-linear function and can be implemented by a deep neural nets, $\mathbf{P}$ and $\mathbf{Q}$ are matrices to map users and items into a joint latent space, respectively. $\mathbf{P}$ and $\mathbf{Q}$ are left out to make the notations more compact when no ambiguity arises. Then, the probability of $c = 1$ is measured by the Sigmoid function:

$$P(c = 1) = \frac{\exp(f(u, v, c = 1))}{1 + \exp(f(u, v, c = 1))} \tag{2}$$

Preliminaries

When an item v is sampled from the real data, we use $y = 1$ or $y = 0$ to represent that user u likes it ($c = 1$) or not ($c = 0$). Similarly, when an item v is from the generator (i.e., fake item), we instead use $y = 3$ or $y = 2$ to represent the same meaning. Next, we briefly introduce how this labeling system supports the two-phase training. In Phase I, since we always train the generator and discriminator under the same class and user (i.e., c is fixed), the label y that discriminator needs to map an input item into is confined to $\{0, 2\}$ when $c = 0$, or $\{1, 3\}$ when $c = 1$, depending on where v is from.

서울시립대학교
UNIVERSITY OF SEOUL

Preliminaries

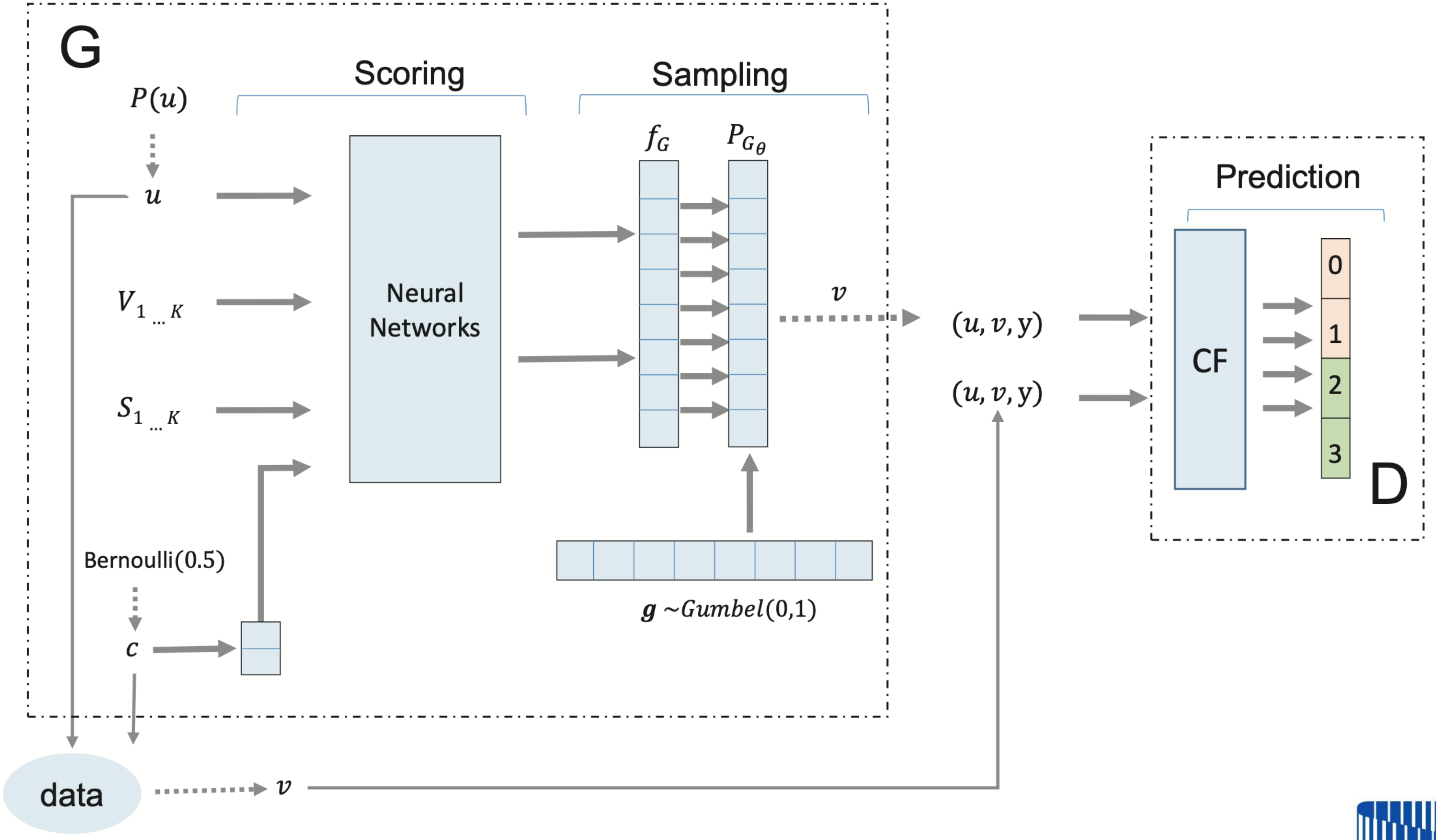
the discriminator just behaves like a normal pure CF method except that it would map a (u, v) pair into both $y = 0$ and $y = 2$ if u dislikes v, otherwise, it is mapped into both $y = 1$ and $y = 3$. As such, we extend the original pure CF score function f introduced above to enable this multi-label objective, in this way, the score for label i can be measured by $f(u, v, y)$ where $y \in \{0, 1, 2, 3\}$, and Sigmoid in Eq. 2 is replaced by Softmax:

$$P(y = i) = \frac{\exp(f(u, v, y = i))}{\sum_{j=0}^{3} \exp(f(u, v, y = j))} \tag{3}$$

Model Overview

AugCF has two major components: the generator (denoted as G_θ) and the discriminator (denoted as D_ϕ) with parameters θ and ϕ respectively. The generator as a data augmentor can generate high-quality and reliable (u, v, c) tuples while the discriminator plays two roles in the model. First, it distinguishes between a real tuple sampled from C and a fake tuple generated by G_θ. Then, it further acts as a pure CF method to predict whether u likes a given v or not.

AugCF



서울시립대학교 UNIVERSITY OF SEOUL

Phase I: Data Augmentation

$$\theta^*, \phi^* = \min_{\theta} \max_{\phi} \Big(\mathbb{E}_{(u,v,y)\sim P_C(v|u,c)} \log[D_\phi(v, y|u, c), y \in \{0, 1\}]$$
$$+ \mathbb{E}_{(u,v,y)\sim P_{G_\theta}(v|u,c)} \log[D_\phi(v, y|u, c), y \in \{2, 3\}] \Big)$$

* $\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[log\ D(x)] + E_{z\sim p_z(z)}[log(1 - D(G(z))]$

서울시립대학교 UNIVERSITY OF SEOUL

Phase I: Data Augmentation

- Discriminator

Given an item under a user and a class, the discriminator needs to discriminate whether it is real or fake. According to Eq. 4, the objective for training the discriminator in Phase I is:

$$\phi_I^* = \max_{\phi}(L_D^I) \tag{5}$$

where

$$L_D^I = \mathbb{E}_{(u,v,y)\sim P_C(v|u,c)} \log[D_\phi(v,y|u,c), y \in \{0,1\}] + \mathbb{E}_{(u,v,y)\sim P_{G_\theta(v|u,c)}} \log[D_\phi(v,y|u,c), y \in \{2,3\}]$$

where $D_\phi(v,y|u,c)$ estimates whether an item is fake or real conditioning on a specific class c and a user u, which is achieved by constraining the probability P to c and u:

$$D_\phi(v,y|u,c) = P(y|v;u,c) = \frac{\exp(f(u,v,y))}{\sum_{i=0}^{3}\exp(f(u,v,y=i))} \tag{6}$$

Phase I: Data Augmentation

- Generator

First, we sample a user u from $\mathcal{U}$ based on this subsampling strategy:

$$P(u) = 1 - \sqrt{\frac{t}{h(u)}} \tag{7}$$

where $h(u)$ is the frequency of user u (i.e., number of u's interactions) and t is a predefined threshold. In this way, the less active users have higher probabilities to be sampled.

Phase I: Data Augmentation

- Generator

We only randomly sample K of them, denoted as $\mathcal{V}_u = \{v_1, \ldots, v_K\}$, and their associated side information is denoted as $\mathcal{S}_u = \{\mathcal{S}_1, \mathcal{S}_1, \ldots, \mathcal{S}_K\}$. Following these steps, u, c, $\mathcal{V}_u$ and $\mathcal{S}_u$ are fed into G_θ, obtaining a point-wise probability distribution $P_{G_\theta(v_i|u,c)}$:

$$P_{G_\theta(v_i|u,c)} = \frac{\exp f_G(u, v_i, \mathcal{S}_i, c)}{\sum_{j=1}^{K} \exp f_G(u, v_j, \mathcal{S}_j, c)} \tag{8}$$

Phase I: Data Augmentation

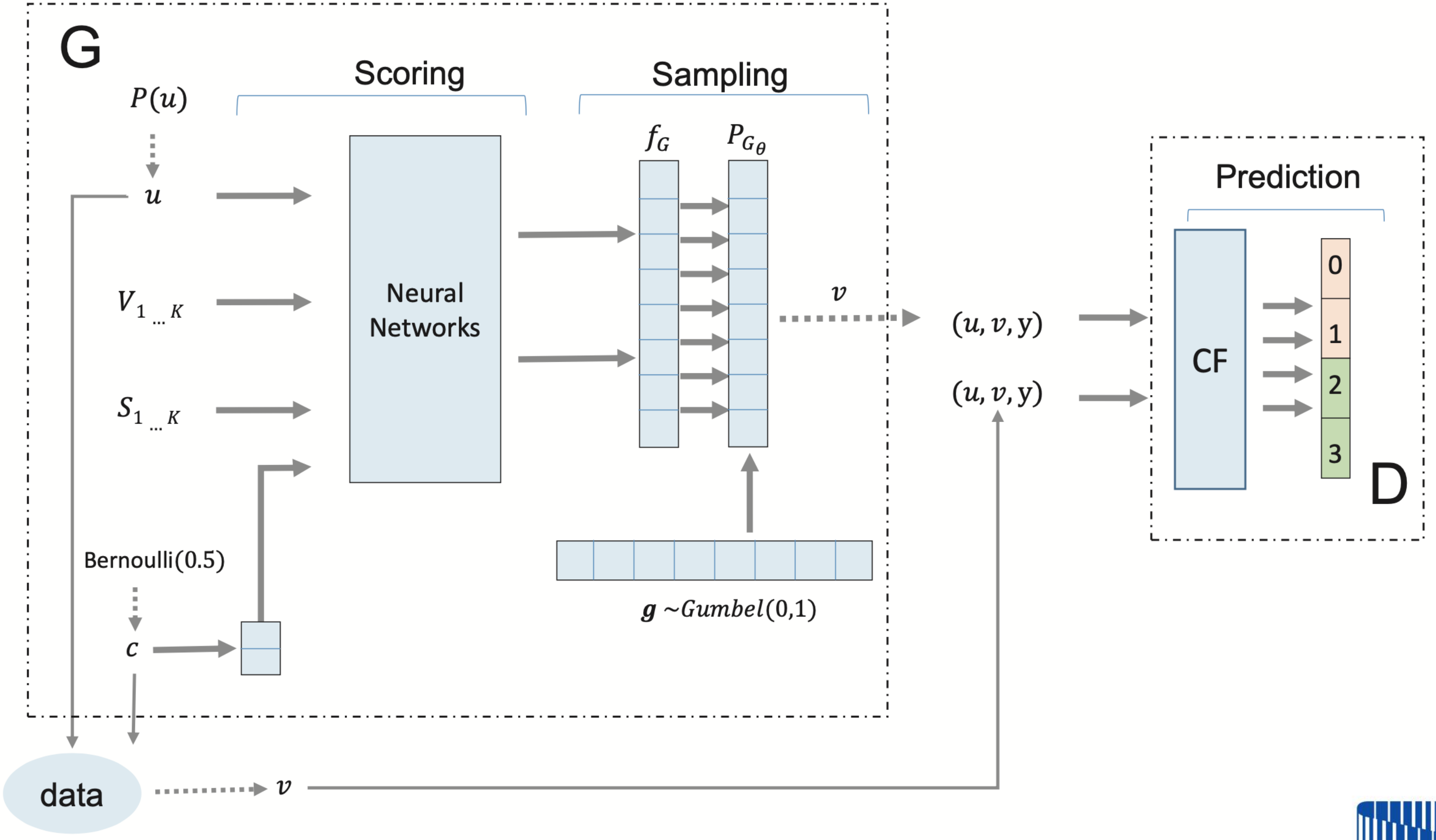
- Generator

$$\theta^* = \min_{\theta}((\mathbb{E}_{(u,v,y)\sim P_C(v|u,c)} \log[D_\phi(v, y|u, c), y \in \{0, 1\}] + \mathbb{E}_{(u,v,y)\sim P_{G_\theta(v|u,c)}} \log[D_\phi(v, y|u, c), y \in \{2, 3\}])$$

$$= \max_{\theta} \underbrace{(\mathbb{E}_{(u,v,y)\sim P_{G_\theta(v|u,c)}} \log[D_\phi(v, y|u, c), y \in \{0, 1\}])}_{\text{denoted as } L_G^I}$$

서울시립대학교 UNIVERSITY OF SEOUL

AugCF



G
Scoring
Sampling
$P(u)$
u
$V_{1 \dots K}$
$S_{1 \dots K}$
Neural Networks
f_G
P_{G_θ}
v
Bernoulli(0.5)
c
$g \sim Gumbel(0,1)$
(u, v, y)
(u, v, y)
Prediction
CF
0
1
2
3
D
data
v

Phase II: Pure Collaborative Filtering

$$\phi^* = \max_{\phi}(\underbrace{\mathcal{L}_R + \mathcal{L}_G}_{\text{denoted as } L_D^{II}}) \tag{10}$$

where,

$$\mathcal{L}_R = \mathbb{E}_{(u,v,y)\sim P_C} \log[D_\phi(y|u,v)+D_\phi(y'|u,v), y \in \{0,1\}, y' \in \{2,3\}]$$

,

$$\mathcal{L}_G = \mathbb{E}_{(u,v,y)\sim P_{G_{\theta^*}}} \log[D_\phi(y|u,v)+D_\phi(y'|u,v), y \in \{2,3\}, y' \in \{0,1\}]$$

and

$$D_\phi(y|u,v) = P(y|u,v) = \frac{\exp(f(u,v,y))}{\sum_{i=0}^{3} \exp(f(u,v,y=i))}$$

End-to-End Training

Specifically, let **g** be a K-dimensional noise vector, where $g_1, \ldots, g_K$ are i.i.d sampled from *Gumbel(0, 1)* [2]. We then obtain the sampled item **v** in one-hot representation, i.e., the position of v in this one-hot vector is 1 while the other elements are 0, using the arg max operation using the Gumbel-Max trick [28]:

$$\mathbf{v} = \text{one\_hot}\left(\arg\max_i \left[\log P_{G_\theta}(v_i) + g_i\right]\right) \tag{11}$$

서울시립대학교
UNIVERSITY OF SEOUL

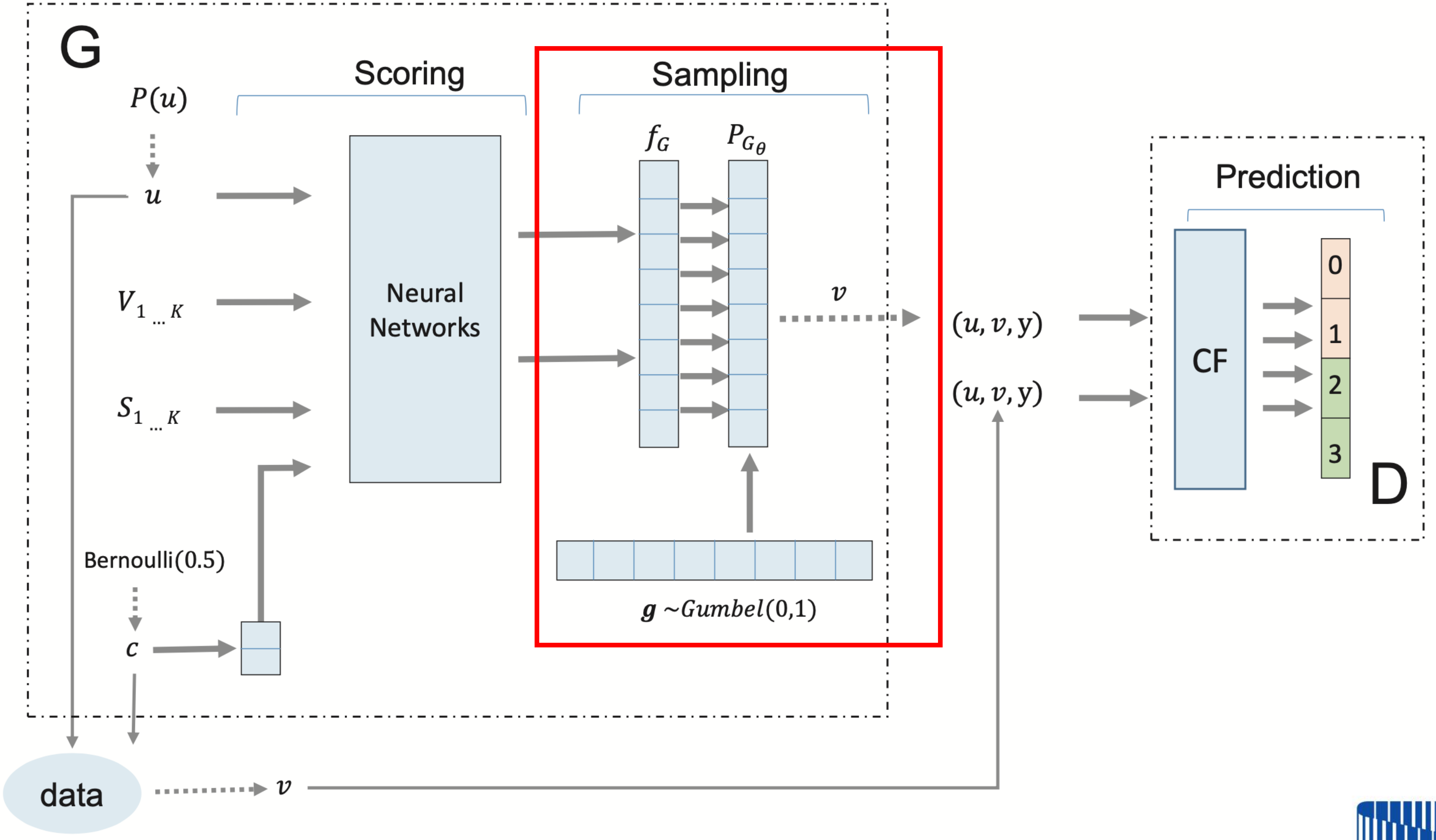
End-to-End Training

Since the arg max operation, again, is not differentiable, the continuous and differentiable softmax function is replaced to approximate it, which is called Gumbel-Softmax [20]. Finally, we obtain an approximate one-hot representation of the sampled item **v**, i.e.,

$$v_i = \frac{\exp((\log P_{G_\theta}(v_i) + g_i)/\tau)}{\sum_{j=1}^{K} \exp((\log P_{G_\theta}(v_j) + g_j)/\tau)} \qquad \text{for } i = 1, \ldots, K \qquad (12)$$

where τ is a hyper-parameter called temperature, and when it approaches 0, samples from the Gumbel-Softmax distribution become one-hot and the Gumbel-Softmax distribution becomes identical to the multinomial distribution P_{G_θ}.

AugCF



서울시립대학교 UNIVERSITY OF SEOUL

Algorithm

Algorithm 1: AugCF Training Algorithm

INPUT: Training data $\mathcal{C}$ and side information $\mathcal{S}$, pre-trained generator G_θ, number of epochs in Phase I and Phase II and batch size J;

OUTPUT: An augmented $\mathcal{C}$ and G_{θ^*} and D_{ϕ^*};

for *each epoch in Phase I* **do**

  $\mathcal{B}_{gen} = \emptyset$; $\mathcal{B}_{real} = \emptyset$; //sets for generated and real data

  Sample a batch of users $u_1, \ldots, u_J$ based on Eq. 7;

  Sample a batch of binary classes $c_1, \ldots, c_J \sim Bern(0.5)$;

  for *every* u, c *in batch* **do**

    G_θ generates a distribution $P_{G_\theta(v|(u,c))}$ over K unvisited items based on Eq. 8;

    Sample a K-dimensional $\mathbf{g} \sim Gumbel(0, 1)$;

    Obtain an approximate fake item v based on Eq. 12;

    Construct a fake tuple (u, v, y) where $y = c + 2$;

    Add this (u, v, y) to $\mathcal{B}_{gen}$;

    Randomly sample an item v from $\mathcal{C}$ given u and c;

    Construct a real tuple (u, v, y) where $y = c$;

    Add this (u, v, y) to $\mathcal{B}_{real}$;

  end

  update G_θ and D_ϕ with $\mathcal{B}_{gen}$ and $\mathcal{B}_{real}$ based on Eq. 13;

end

for *each epoch in Phase II* **do**

  $\mathcal{B}_{gen} = \emptyset$; $\mathcal{B}_{real} = \emptyset$; //sets for generated and real data

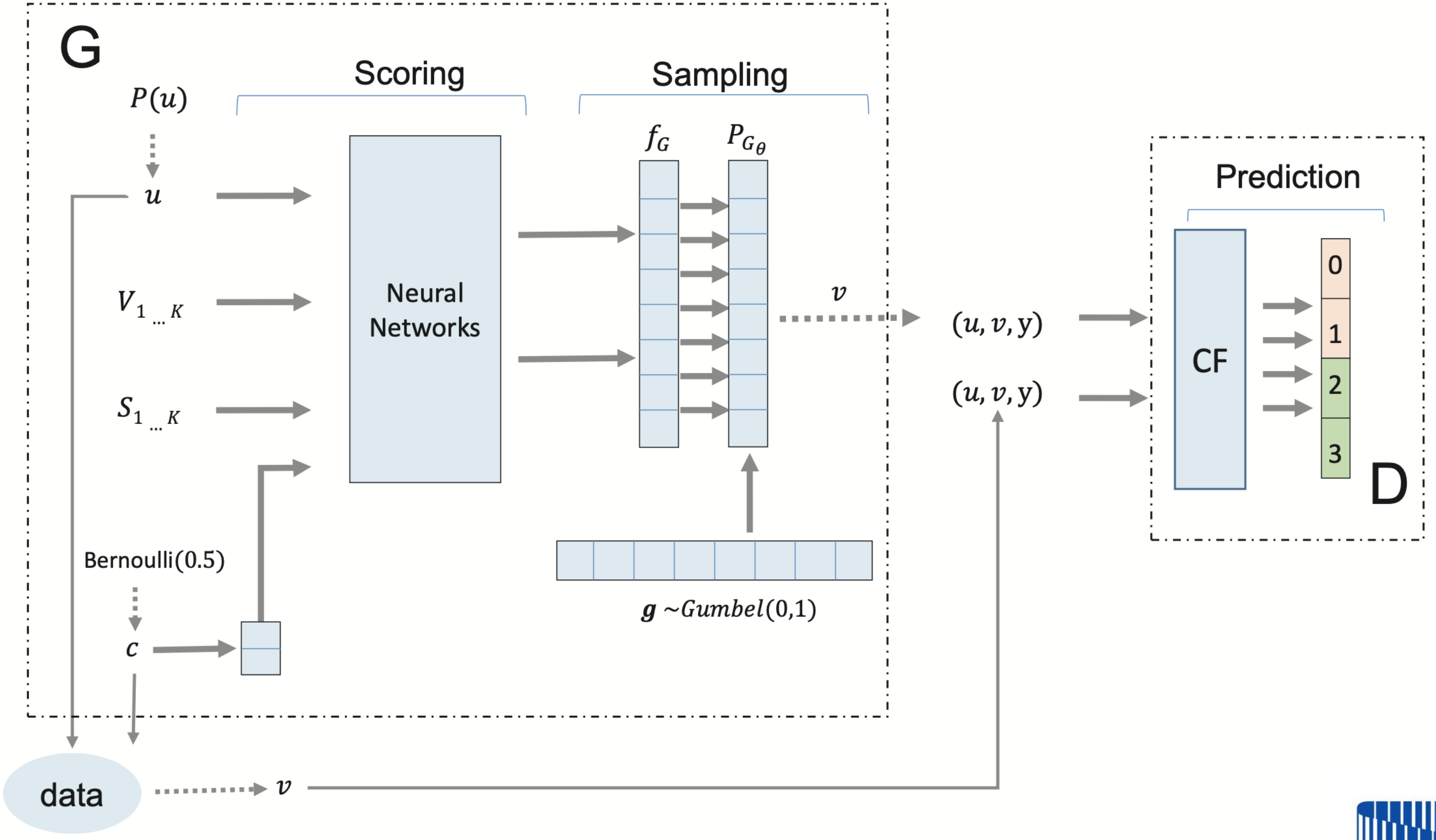
  G_{θ^*} generates a batch of tuples, adding to $\mathcal{B}_{gen}$;

  Add a batch of tuples sampled from $\mathcal{C}$ to $\mathcal{B}_{real}$;

  Update D_ϕ with both $\mathcal{B}_{gen}$ and $\mathcal{B}_{real}$ based on Eq. 14;

end

AugCF



서울시립대학교
UNIVERSITY OF SEOUL

Applications

❑ Content-Based AugCF (DeepCoNN)

$$f_G(u, v_i, \mathcal{S}_i, c) = \hat{w}_0 + \sum_{i=1}^{|\hat{z}|} \hat{w}_i \hat{z}_i + \sum_{i=1}^{|\hat{z}|} \sum_{j=i+1}^{|\hat{z}|} \left\langle \hat{\mathbf{b}}_i, \hat{\mathbf{b}}_j \right\rangle \hat{z}_i \hat{z}_j$$

❑ Sparse Feature-Based AugCF (Wide & Deep)

$$f_G(u, v_i, \mathcal{S}_i, c) = \mathbf{w}_{wide}^T [\mathbf{x}, \phi(\mathbf{x})] + \mathbf{w}_{deep}^T a^{(l_f)} + b$$

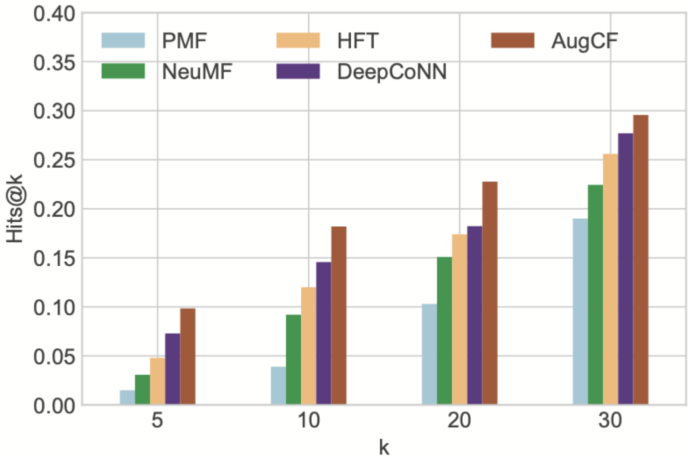
Datasets

Content-Based

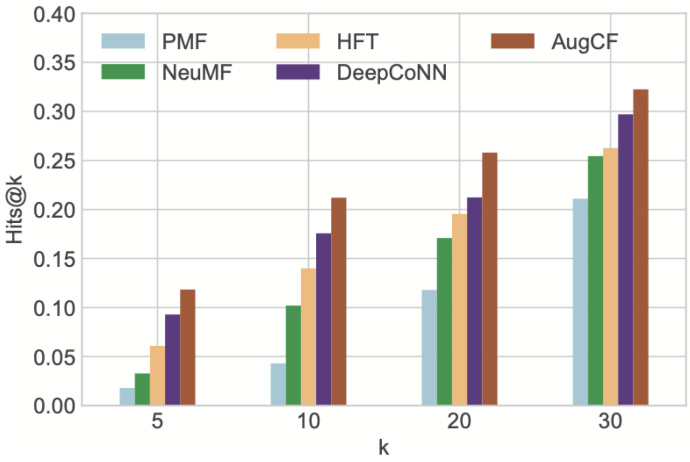
Dataset	#Users	#Items	#Interactions	#Features	Sparsity
Electronics	192,403	63,001	1,689,188	N/A	0.014%
Health	38,609	18,534	346,355	N/A	0.048%
Beauty	22,363	12,101	198,502	N/A	0.073%
Games	24,303	10,672	231,780	N/A	0.089%
Movielens	6,040	3,706	1,000,209	9,812	4.47%
Frappe	957	4,082	288,609	5,382	7.39%

Sparse Feature-Based

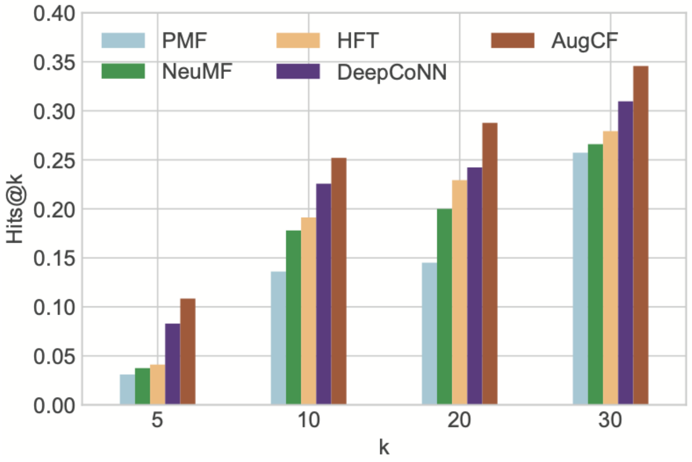
Experiments



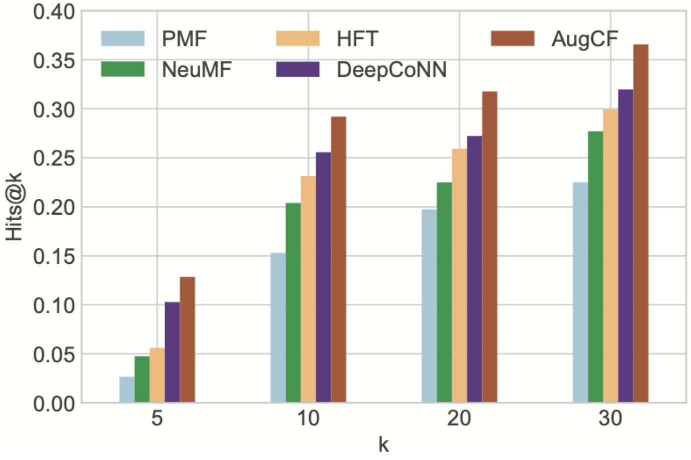
(a) Electronics



(b) Health

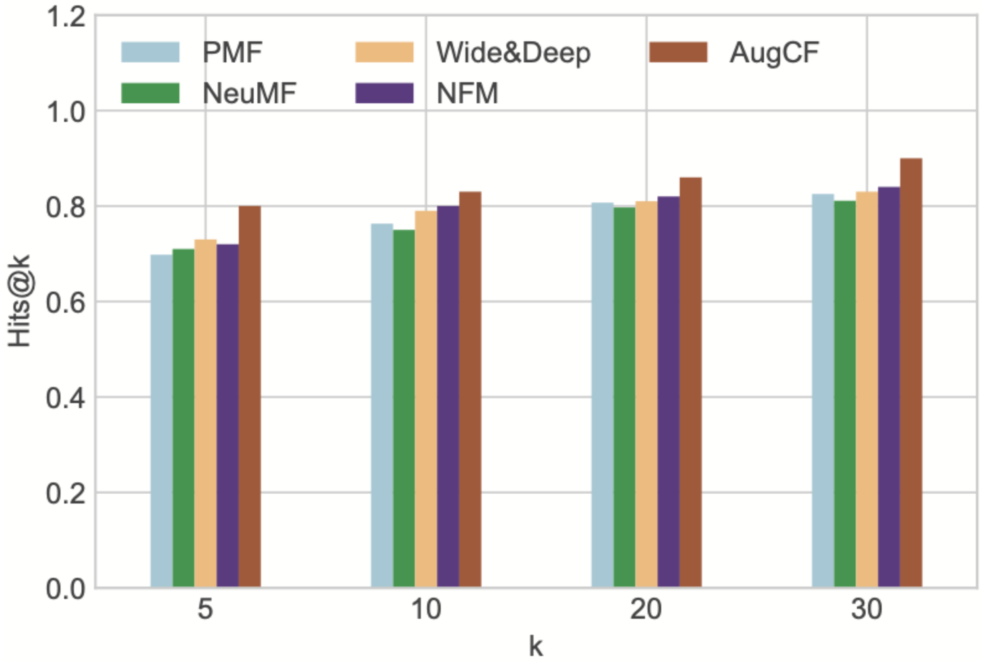


(c) Beauty

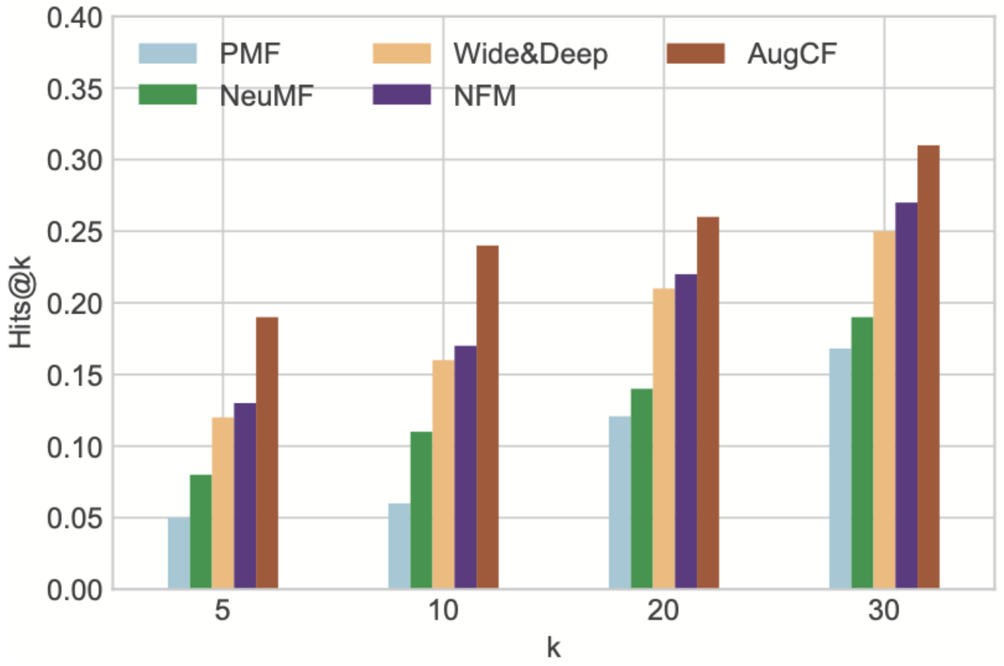


(d) Games

Experiments



(a) Frappe



(b) Movielens