13기추천세미나

ToBig's 13기 조상연

Learning (to Rank and Beyond)

Contents

Unit 01 | Learning Recommenders

Unit 02 | Netflix & BPR

Unit 03 | Multi Armed Bandits

Unit 04 | Deep Learning Based Recommendation

TL; DR

1. Recommender 학습은 우리가 잘 아는 것 처럼 Loss Function 잘 설정하고 이에 맞게 Gradient Descent 같은 걸 통해 최적의 해를 찾아가는 것이다.

2. BPR (베이지안 개인화 랭킹) 은 AUC를 최대화 하는 것에 목적이 있으며 로지스틱을 통해 학습이 가능하도록 만들었다.

3. Multi armed Bandit은 기존 A/B Testing의 한계를 극복하며 더 효율적이고 안전한 테스팅을 할 수 있게 해준다.

4. 최근의 Recommendation System에선 딥러닝 기반의 모델이 많이 나오고 있는 추세이며 오토인코더, MLP, RNN, Attention 등 여러 기법이 적용되고 있다.

Unit 01 | Learning Recommenders

Structure of Learning

$$\underset{\Theta}{\operatorname{argmin}}\, g(\Theta)$$

- Θ: parameters and/or recommendation model
- $g(\Theta)$: error or utility computing predictions or recommendations with model Θ
  - when using utility, maximize instead of minimize

- A *model*, such as matrix factorization or linear regression
  - Model generally has *parameters* (Θ)
- A *utility function* or *loss function*, measuring how good or bad output is
- An *optimization algorithm* to find best parameters

1. Recommender 학습은 우리가 아는 머신러닝과 크게 유사하다.

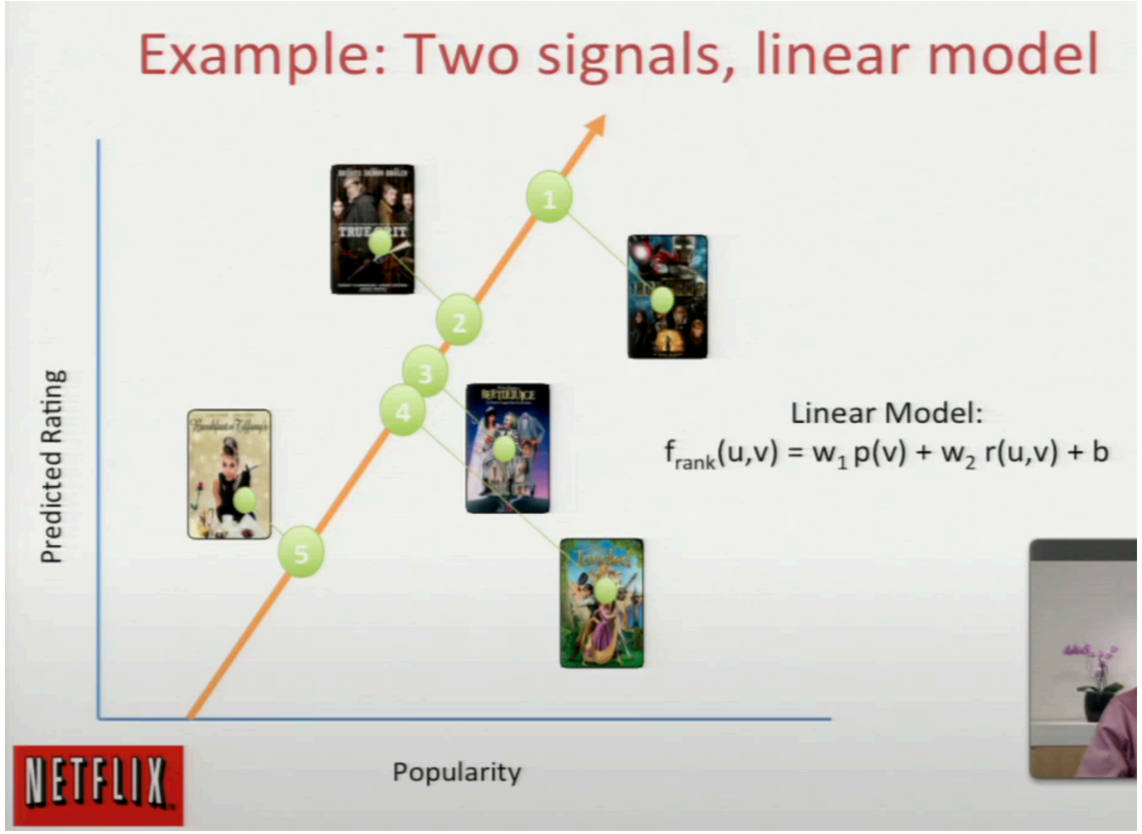
2. 크게 Model / Loss function / Optimization 으로 구성되어 있음

3. Model: Linear model, Funk SVD
   Loss Function: RMSE, Accuracy
   Optimization: Gradient Descent

Netflix Recommendation

- 단순히 클릭 유도 뿐 아니라 오래 볼 수 있는 콘텐츠를 추천할 수 있도록 최적화를 진행
- 예전엔 피드백의 퀄리티가 좋았지만 요즘엔 아님, 잠재적 피드백은 무궁무진하기에 이걸 잘 차용
- **Rating 을 높게 주는 것을 예측하는 것이 아님**

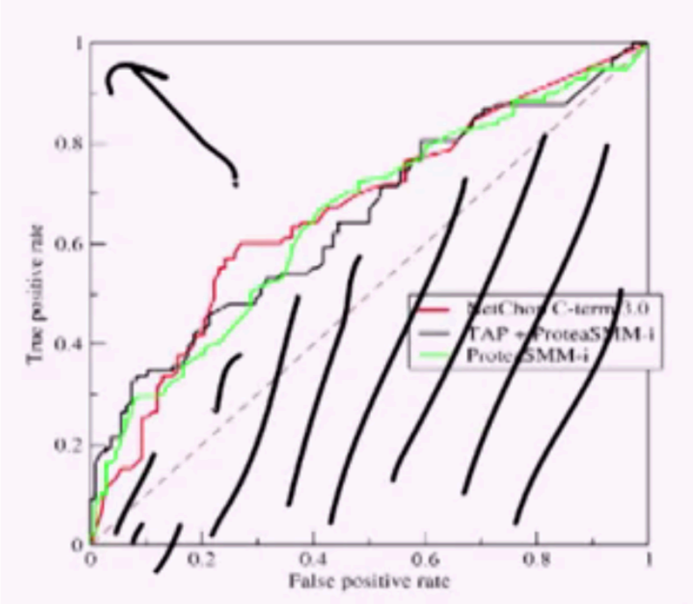
Unit 02 | Netflix & BPR



BPR (Bayesian Personalized Ranking)

= Model + Loss Function + Optimization

= MF + BPR – OPT (AUC) - SGD



$$AUC(D) = \frac{\sum_{(u,i,j)\in D} 1\{S(i;u) > S(j;u)\}}{|D|}$$

- AUC 자체는 학습이 불가

$$AUC(D) \approx \sum_{(u,i,j)\in D} 1\{S(i;u) > S(j;u)\}$$

$$AUC(D) \approx \sum_{(u,i,j)\in D} log(\sigma(S(i;u) - S(j;u)))$$

$$AUC \approx \text{BPR-OPT}$$

- 이를 Logistic function 을 통해 학습 가능하게 바꾸고 log를 취해 더 빠른 학습을 가능하게 함

BPR (Bayesian Personalized Ranking)

1. 데이터 준비
    1. 유저, 아이템(O), 아이템(X)
    2. 유저가 행위를 한 / 하지 않은 아이템을 학습
2. 데이터 순서
    1. **랜덤** / 처음부터 끝까지 다 하지는 말 것

```python
r_uij = np.sum(user_u * (item_i - item_j), axis = 1)
sigmoid = np.exp(-r_uij) / (1.0 + np.exp(-r_uij))

# repeat the 1 dimension sigmoid n_factors times so
# the dimension will match when doing the update
sigmoid_tiled = np.tile(sigmoid, (self.n_factors, 1)).T

# update using gradient descent
grad_u = sigmoid_tiled * (item_j - item_i) + self.reg * user_u
grad_i = sigmoid_tiled * -user_u + self.reg * item_i
grad_j = sigmoid_tiled * user_u + self.reg * item_j
self.user_factors[u] -= self.learning_rate * grad_u
self.item_factors[i] -= self.learning_rate * grad_i
self.item_factors[j] -= self.learning_rate * grad_j
```

http://ethen8181.github.io/machine-learning/recsys/4_bpr.html

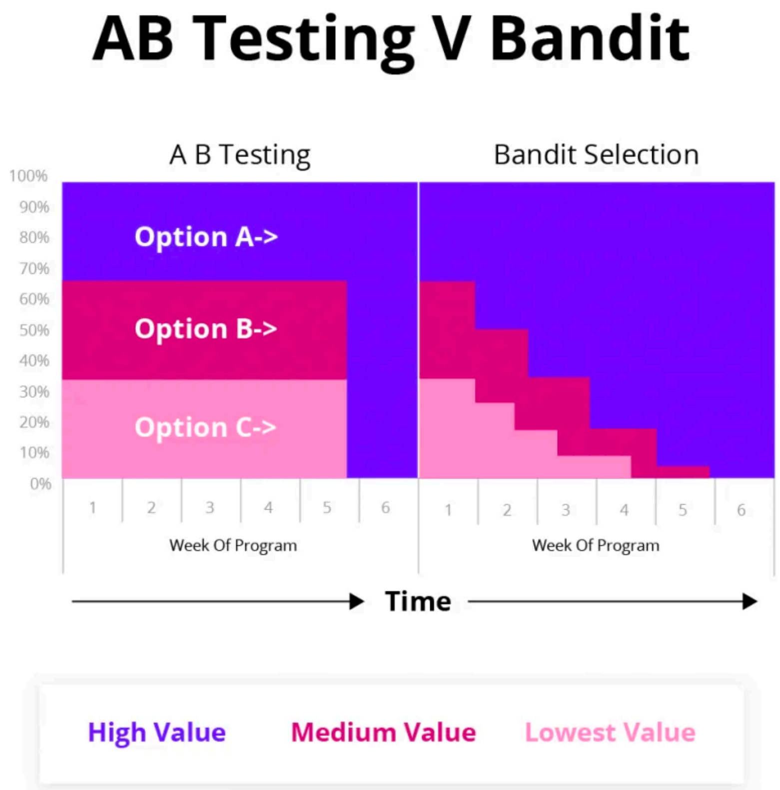
Multi Armed Bandit

Unit 03 | Multi Armed Bandit

내가 다음에 어떤 슬롯머신을 당겨야 할까

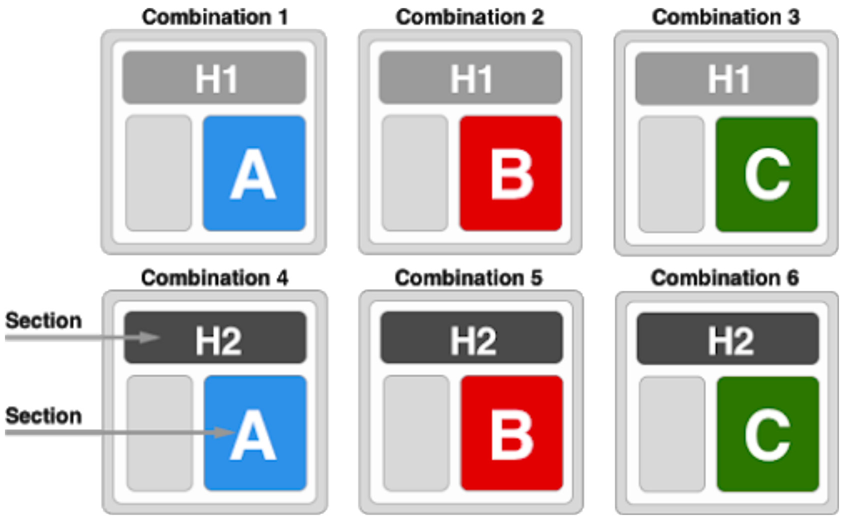
Exploration and Exploitation 을 통해 해결

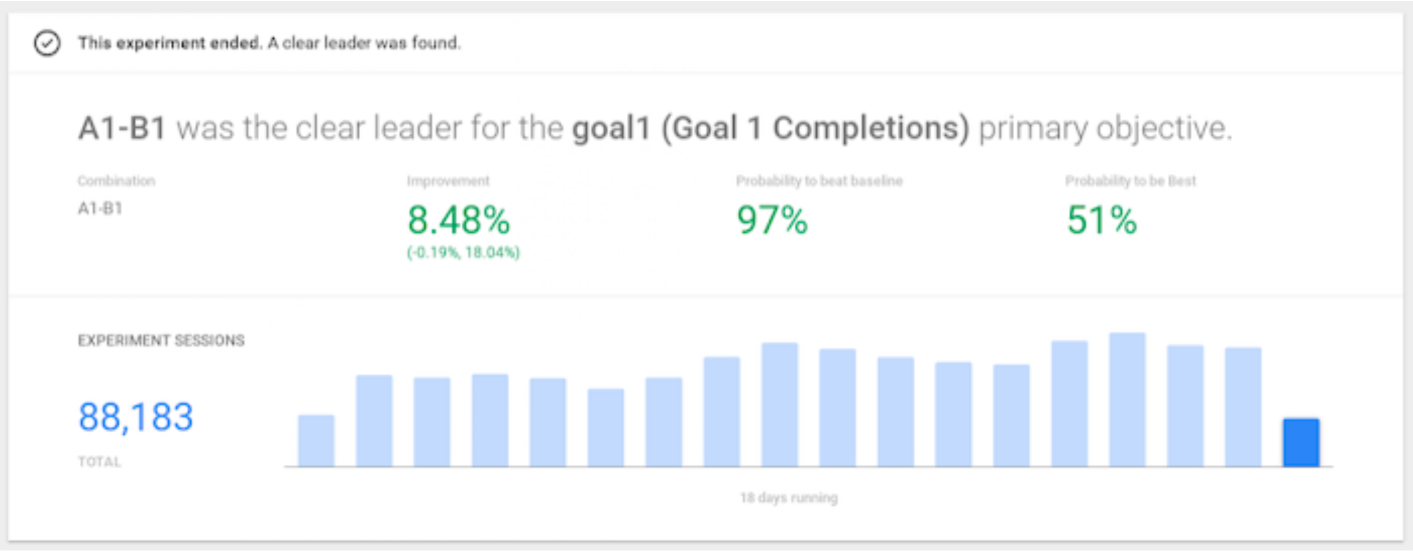
MAB vs A/B Testing



1. Saving Time
2. User Confidence
3. Less Regret, Less Cost
4. Multivariate Testing

Multivariate Testing





Google Optimize

MAB 를 위한 여러 알고리즘

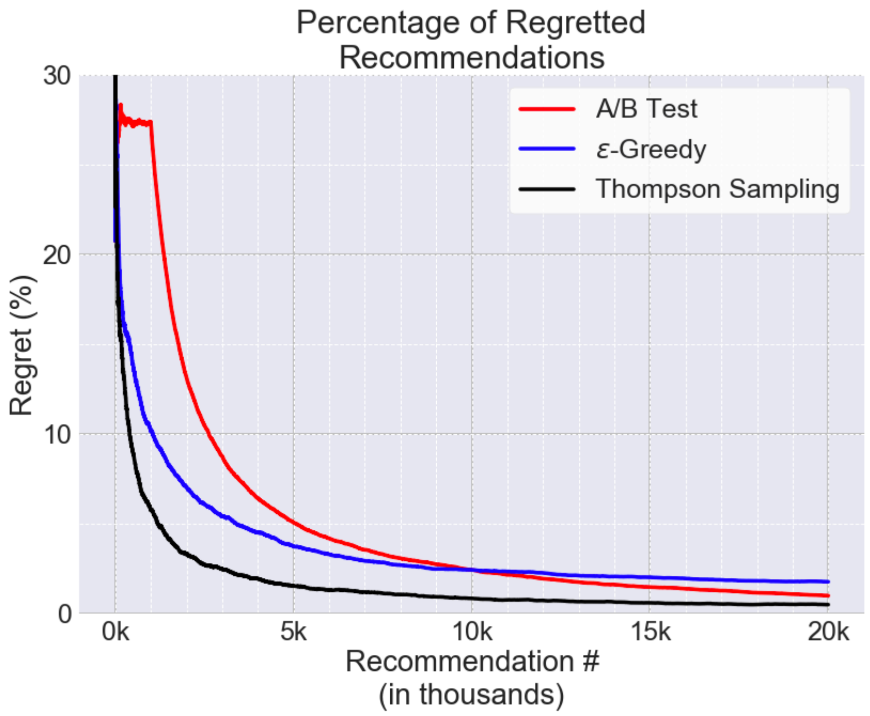
1. Greedy, e-Greedy 알고리즘

$$\boxed{Q_t(a)} \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot \mathbf{1}_{A_i=a}}{\sum_{i=1}^{t-1} \mathbf{1}_{A_i=a}}$$

2. UCB (Upper Confidence Bound)

$$A_t \doteq \underset{a}{\arg\max} \left[\boxed{Q_t(a)} + c\sqrt{\frac{\log t}{N_t(a)}} \right]$$

3. Thompson Sampling

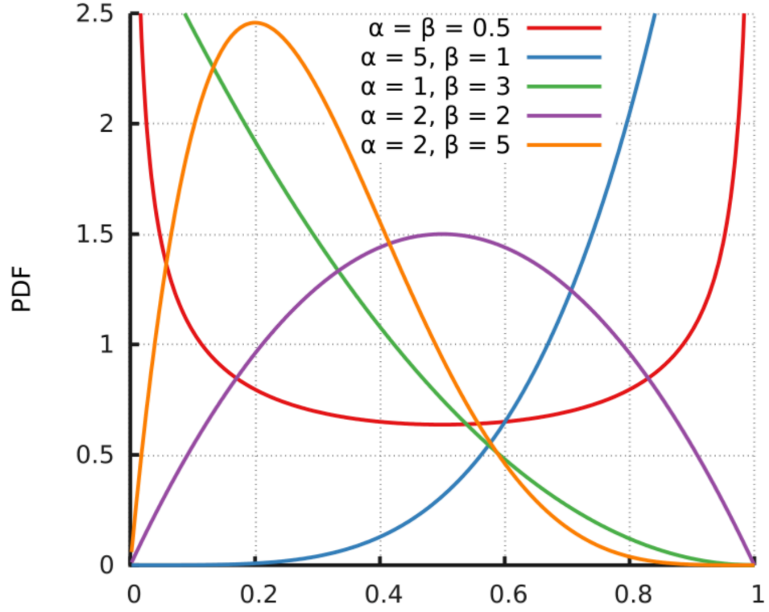


https://brunch.co.kr/@chris-song/62

Thompson Sampling

$$Beta(x|a,b) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}$$

베타 분포



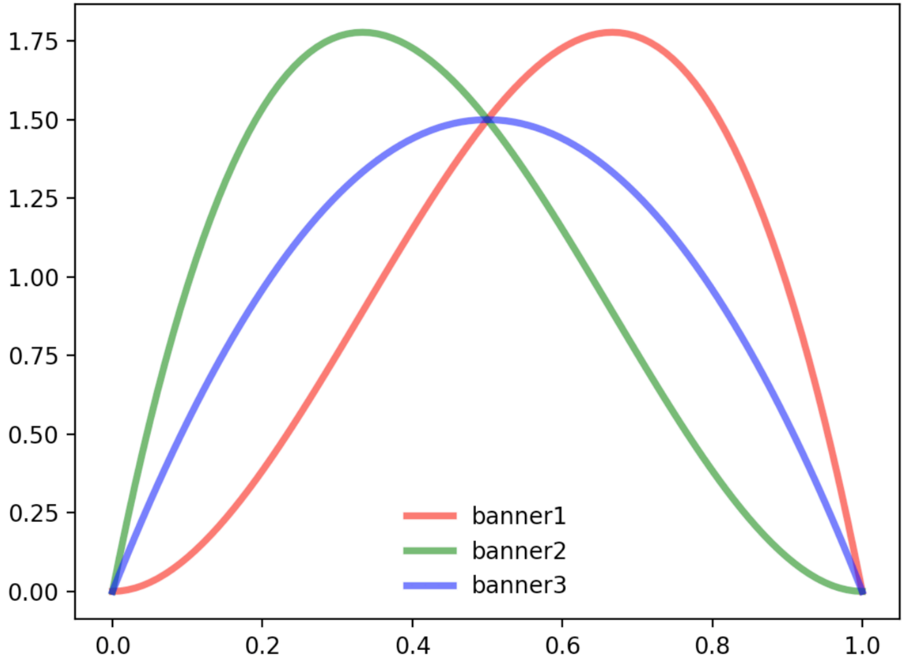
알파와 베타 값에 따른 베타 분포 출처: 위키피디아

https://brunch.co.kr/@chris-song/66

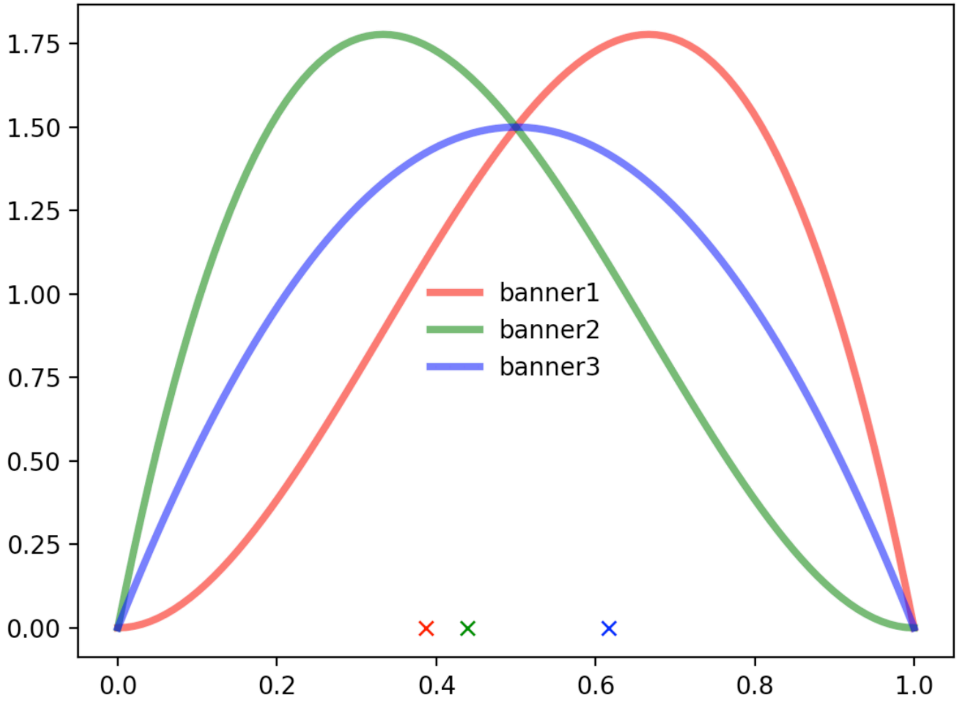
Unit 03 | Multi Armed Bandit

Beta(클릭한 횟수+1, 클릭하지 않은 횟수+1)

banner1: Beta(3, 2)
banner2: Beta(2, 3)
banner3: Beta(2, 2)



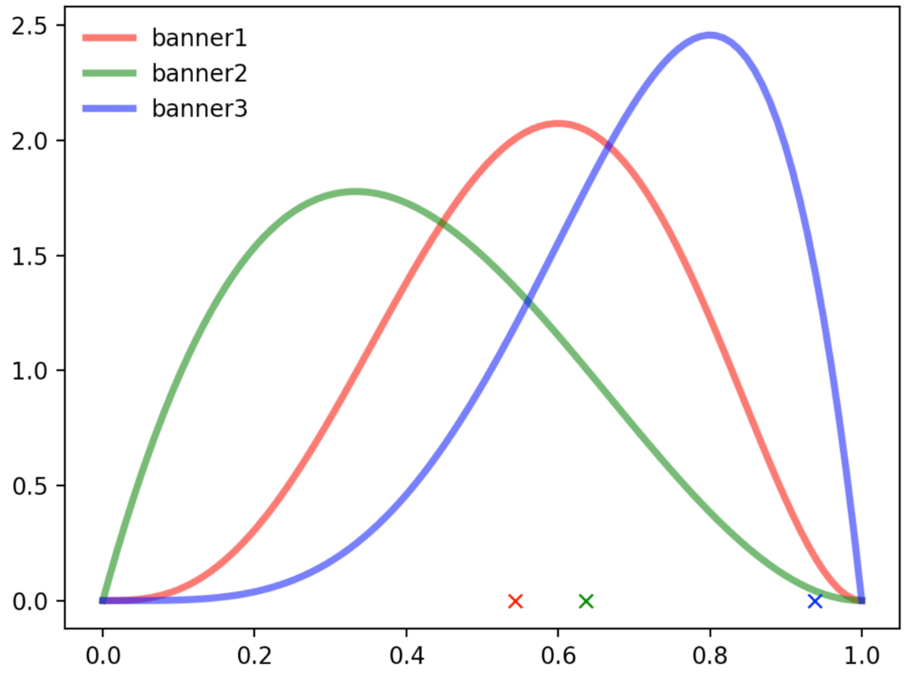
샘플링

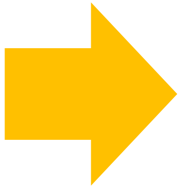


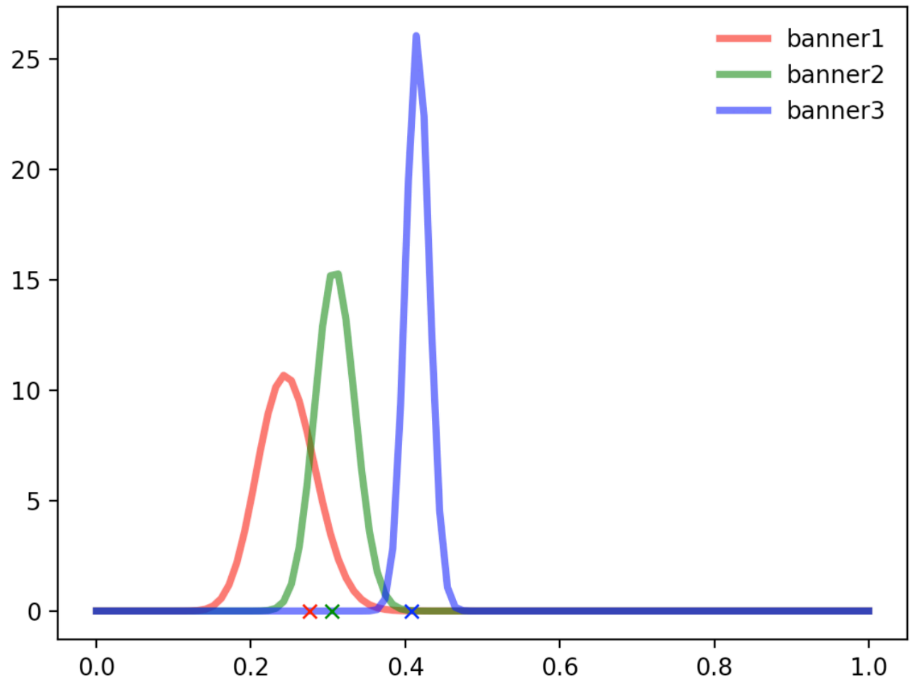
https://brunch.co.kr/@chris-song/66

Beta(클릭한 횟수+1, 클릭하지 않은 횟수+1)

banner1: Beta(4, 3)
banner2: Beta(2, 3)
banner3: Beta(5, 2)

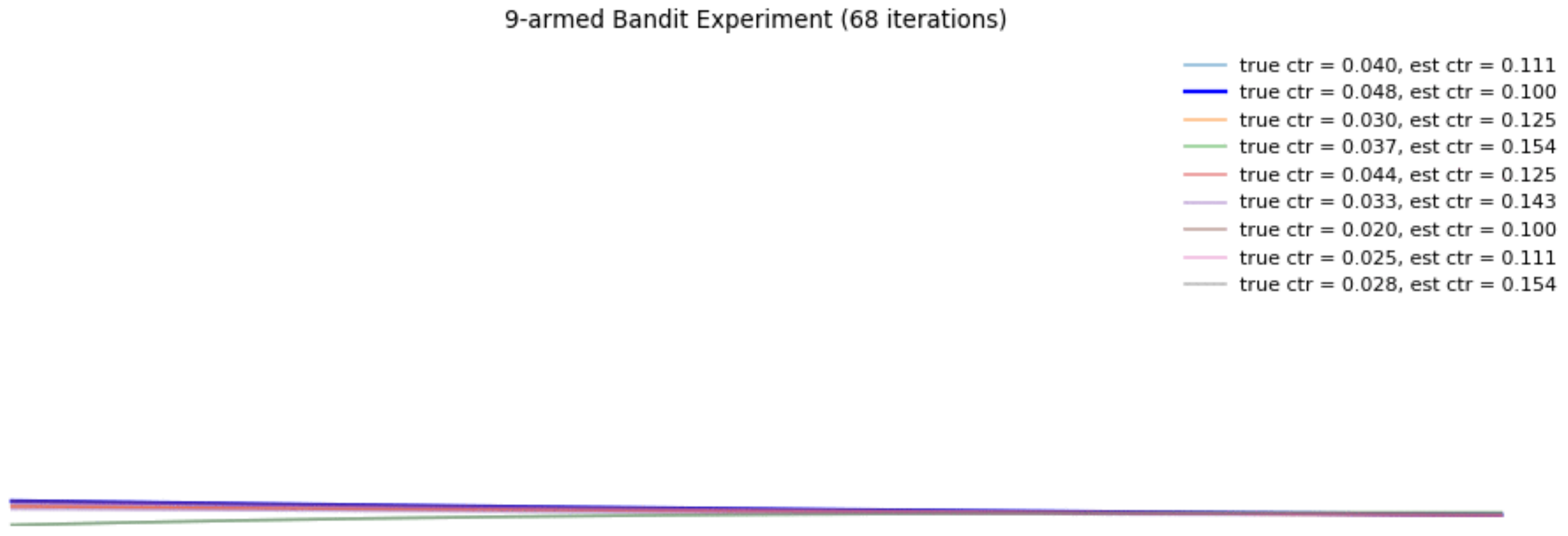






https://brunch.co.kr/@chris-song/66

Thompson Sampling



https://towardsdatascience.com/beyond-a-b-testing-multi-armed-bandit-experiments-1493f709f804

Unit 03 | Multi Armed Bandit

1. 단순한 클릭 뿐 아니라 다양한 행위 (구매, 시청) 등 적용 가능
2. **Context 정보**를 넣거나, **시간에 따라 decay** 하도록 하는 등의 기법 적용하는 추세
3. 다음과 같은 상황에서 좋다고 함
    1. 테스트로 인한 **손실이 매우 큰 경우**
    2. 스타트업과 같이 **트래픽이 적고 귀한 경우**
    3. **여러가지 변수에 대해 테스트** 하고 싶을 경우

Deep Learning Based Recommendation

Unit 04 | Deep Learning Based Recommendation

Deep Learning Based Recommender System: A Survey and New Perspectives

SHUAI ZHANG and LINA YAO, University of New South Wales
AIXIN SUN and YI TAY, Nanyang Technological University

With the growing volume of online information, recommender systems have been an effective strategy to overcome information overload. The utility of recommender systems cannot be overstated, given their widespread adoption in many web applications, along with their potential impact to ameliorate many problems related to over-choice. In recent years, deep learning has garnered considerable interest in many research fields such as computer vision and natural language processing, owing not only to stellar performance but also to the attractive property of learning feature representations from scratch. The influence of deep learning is also pervasive, recently demonstrating its effectiveness when applied to information retrieval and recommender systems research. The field of deep learning in recommender system is flourishing. This article aims to provide a comprehensive review of recent research efforts on deep learning-based recommender systems. More concretely, we provide and devise a taxonomy of deep learning-based recommendation models, along with a comprehensive summary of the state of the art. Finally, we expand on current trends and provide new perspectives pertaining to this new and exciting development of the field.

CCS Concepts: • **Information systems** → **Recommender systems**;

Additional Key Words and Phrases: Recommender system, deep learning, survey

ACM Reference format:
Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1, Article 5 (February 2019), 38 pages.
https://doi.org/10.1145/3285029

Netflix 80%

Youtube 60%

State-of-the-Art

https://dl.acm.org/doi/pdf/10.1145/3285029

Unit 04 | Deep Learning Based Recommendation

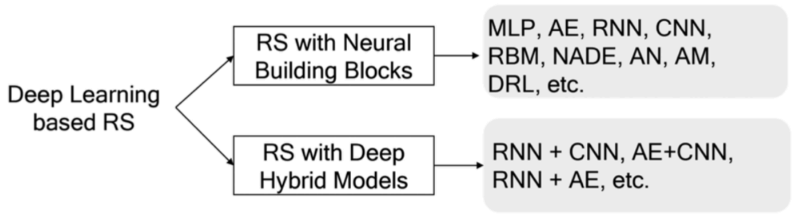


Fig. 1. Categories of deep neural network-based recommendation models.

Table 1. Publications Based on Different Deep Learning Techniques

Categories	Publications
MLP	[2, 13, 20, 27, 38, 47, 53, 54, 66, 92, 95, 158, 167, 186], [12, 39, 93, 112, 135, 155, 183, 184]
Autoencoder	[34, 88, 89, 114, 116, 126, 137, 138, 141, 160, 178, 188, 208], [4, 10, 32, 94, 151, 152, 159, 171, 172, 189, 197, 209, 210]
CNNs	[25, 49, 50, 75, 76, 99, 105, 128, 131, 154, 166, 173, 203, 207], [6, 44, 51, 83, 110, 127, 144, 149, 170, 191, 192]
RNNs	[5, 28, 35, 56, 57, 73, 78, 90, 118, 133, 140, 143, 175–177], [24, 29, 33, 55, 68, 91, 108, 113, 134, 142, 150, 174, 180]
RBM	[42, 71, 72, 101, 124, 168, 181]
NADE	[36, 204, 205]
Neural Attention	[14, 44, 70, 90, 100, 103, 128, 146, 170, 190, 195, 206], [62, 147, 194]
Adversary Network	[9, 52, 163, 165]
DRL	[16, 21, 107, 169, 199–201]
Hybrid Models	[17, 38, 41, 82, 84, 87, 119, 136, 161, 193, 194]

Table 2. Deep Neural Network-Based Recommendation Models in Specific Application Fields

Data Sources/Tasks	Notes	Publications
Sequential Information	w/t User ID	[16, 29, 33, 35, 73, 91, 118, 134, 144, 161, 174, 176, 190, 195, 199, 206]
	Session based w/o User ID	[55–57, 68, 73, 100, 102, 103, 118, 143, 149, 150]
	Check-In, POI	[151, 152, 166, 186]
Text	Hash Tags	[44, 110, 119, 159, 183, 184, 194, 210]
	News	[10, 12, 113, 136, 170, 201]
	Review texts	[11, 87, 127, 147, 175, 198, 203]
	Quotes	[82, 142]
Images	Visual features	[2, 14, 25, 49, 50, 84, 99, 105, 112, 166, 173, 180, 192, 193, 198, 207]
Audio	Music	[95, 154, 168, 169]
Video	Videos	[14, 17, 27, 83]
Networks	Citation Network	[9, 38, 66]
	Social Network	[32, 116, 167]
	Cross Domain	[39, 92, 167]
Others	Cold-start	[155, 157, 171, 172]
	Multitask	[5, 73, 87, 175, 188]
	Explainability	[87, 127]

Unit 04 | Deep Learning Based Recommendation

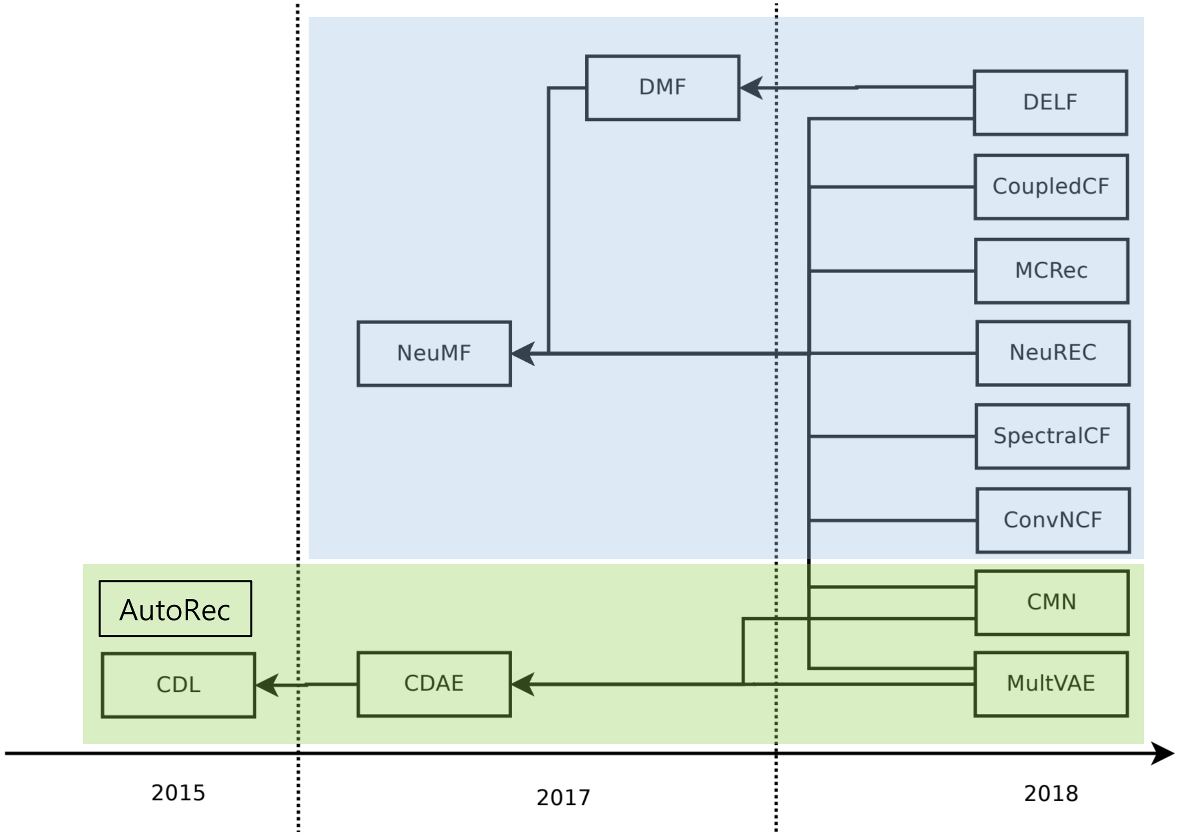
task cate	w/t User ID	Session based w/o User ID	Check-In, POI	Hash Tags	News	Review texts	Quotes	Visual features	Music	Videos	Citation Network	Social Network	Cross Domain	Cold-start	Multitask	Explainability
Adversary Network											[9.0]					
Autoencoder			[151.0, 152.0]	[159.0, 210.0]	[10.0]							[32.0, 116.0]		[171.0, 172.0]	[188.0]	
CNNs	[144.0]	[149.0]	[166.0]	[44.0, 110.0]	[170.0]	[127.0, 203.0]		[166.0, 25.0, 49.0, 50.0, 99.0, 105.0, 173.0, ...	[154.0]	[83.0]						[127.0]
DRL	[16.0, 199.0]				[201.0]				[169.0]							
Hybrid Models	[161.0]			[119.0, 194.0]	[136.0]	[87.0]	[82.0]	[84.0, 193.0]		[17.0]	[38.0]				[87.0]	[87.0]
MLP			[186.0]	[183.0, 184.0]	[12.0]			[2.0, 112.0]	[95.0]	[27.0]	[38.0, 66.0]	[167.0]	[167.0, 39.0, 92.0]	[155.0]		
Neural Attention	[190.0, 195.0, 206.0]	[100.0, 103.0]		[44.0, 194.0]	[170.0]	[147.0]		[14.0]		[14.0]						
RBM									[168.0]							
RNNs	[29.0, 33.0, 35.0, 73.0, 91.0, 118.0, 134.0, 1...	[73.0, 118.0, 55.0, 56.0, 57.0, 68.0, 143.0, 1...			[113.0]	[175.0]	[142.0]	[180.0]							[73.0, 175.0, 5.0]	

1.MLP 기반 모델

2.Autoencoder 기반 모델

3.Session Based Recommendation

Fig. 1. Overview of Neural Methods, arrows indicate when a newer method used another one as baseline in the experiments.



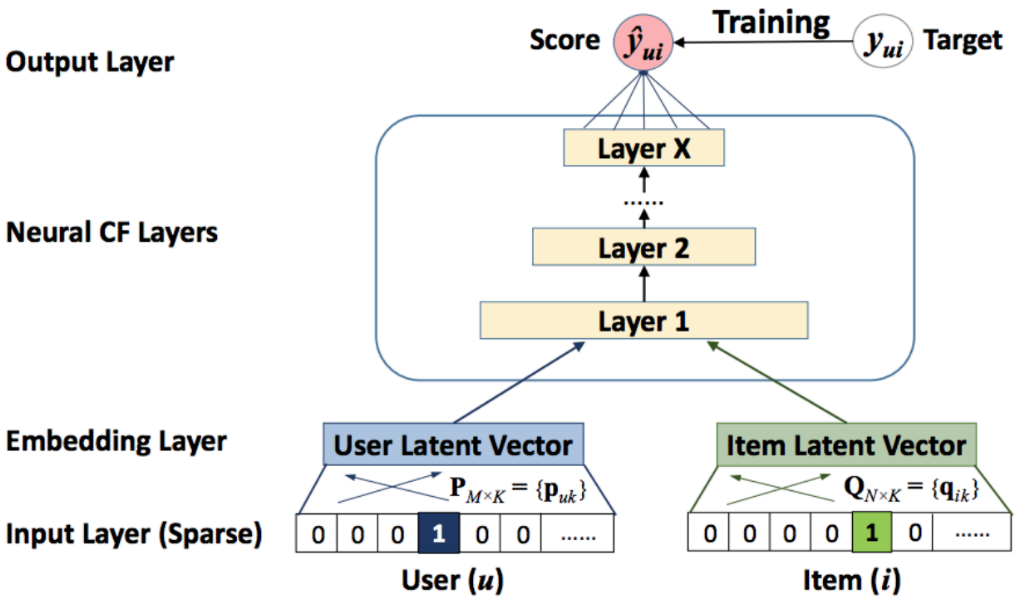
1. MLP 기반 모델

2. Autoencoder 기반 모델

Dacrema, Maurizio Ferrari, et al. "A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research." *arXiv preprint arXiv:1911.07698* (2019).

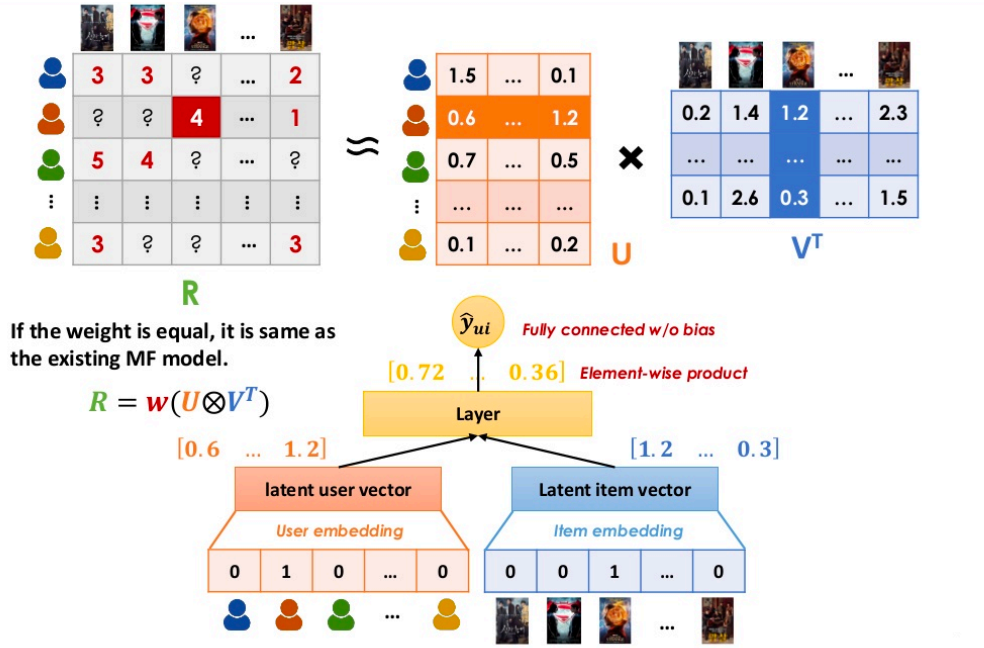
1.MLP 기반 모델: NeulMF

1. Generalized Matrix Factorization (GMF)
2. Multi-Layer Perceptron (MLP)
3. NeulMF: Fusion of GMF and MLP



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

1.1 Generalized Matrix Factorization (GMF)

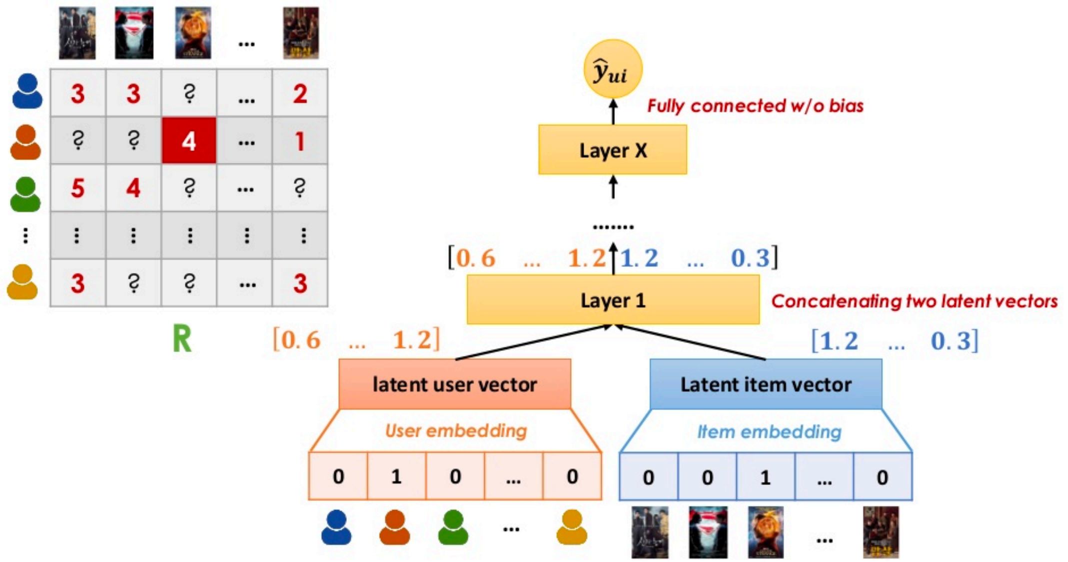



```
# Input variables
user_input = Input(shape=(1,), dtype='int32', name = 'user_input')
item_input = Input(shape=(1,), dtype='int32', name = 'item_input')

 MF_Embedding_User = Embedding(input_dim = num_users, output_dim = latent_dim, name = 'user_embedding',
MF_Embedding_User = Embedding(input_dim = num_users, output_dim = latent_dim, name = 'user_embedding', e
 MF_Embedding_Item = Embedding(input_dim = num_items, output_dim = latent_dim, name = 'item_embedding',
MF_Embedding_Item = Embedding(input_dim = num_items, output_dim = latent_dim, name = 'item_embedding', e

# Crucial to flatten an embedding vector!
user_latent = Flatten()(MF_Embedding_User(user_input))
item_latent = Flatten()(MF_Embedding_Item(item_input))

# Element-wise product of user and item embeddings
predict_vector = merge([user_latent, item_latent], mode = 'mul')

# Final prediction layer
#prediction = Lambda(lambda x: K.sigmoid(K.sum(x)), output_shape=(1,))(predict_vector)
prediction = Dense(1, activation='sigmoid', init='lecun_uniform', name = 'prediction')(predict_vector)

model = Model(input=[user_input, item_input],
            output=prediction)
```

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

1.2 Multi-Layer Perceptron (MLP)

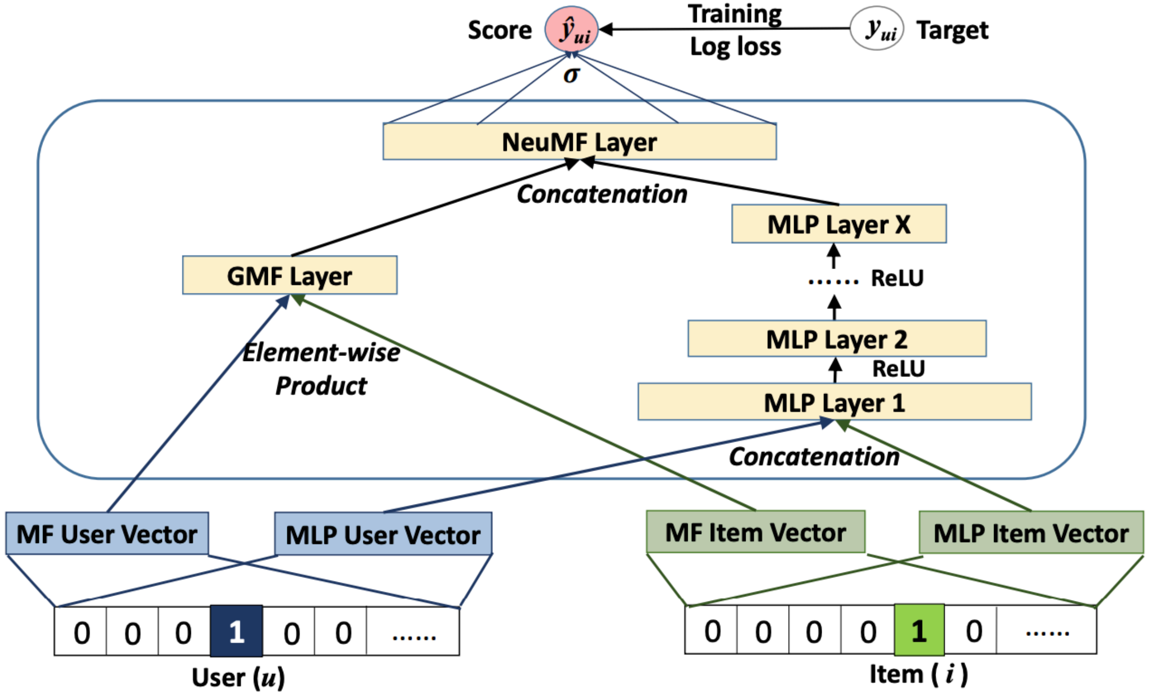
➤ **Learning non-trivial interactions between users and items**




```
# Crucial to flatten an embedding vector!
user_latent = Flatten()(MLP_Embedding_User(user_input))
item_latent = Flatten()(MLP_Embedding_Item(item_input))

# The 0-th layer is the concatenation of embedding layers
vector = merge([user_latent, item_latent], mode = 'concat')

# MLP layers
for idx in xrange(1, num_layer):
    layer = Dense(layers[idx], W_regularizer= l2(reg_layers[idx]), activation='relu', name = 'layer%d' %idx)
    vector = layer(vector)

# Final prediction layer
prediction = Dense(1, activation='sigmoid', init='lecun_uniform', name = 'prediction')(vector)
```

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

1.3 NeuMF: Fusion of GMF and MLP



RQ1 Do our proposed NCF methods outperform the state of-the-art implicit collaborative filtering methods?

A. 대부분 지표에서 더 좋은 성능을 보인다. 특히 Pre-trained 중요

RQ2 How does our proposed optimization framework (log loss with negative sampling) work for the recommendation task?

A. 네거티브 샘플링을 통해 더 강건한 모델을 만들 수 있지만 너무 과하면 역시 좋지 않다.

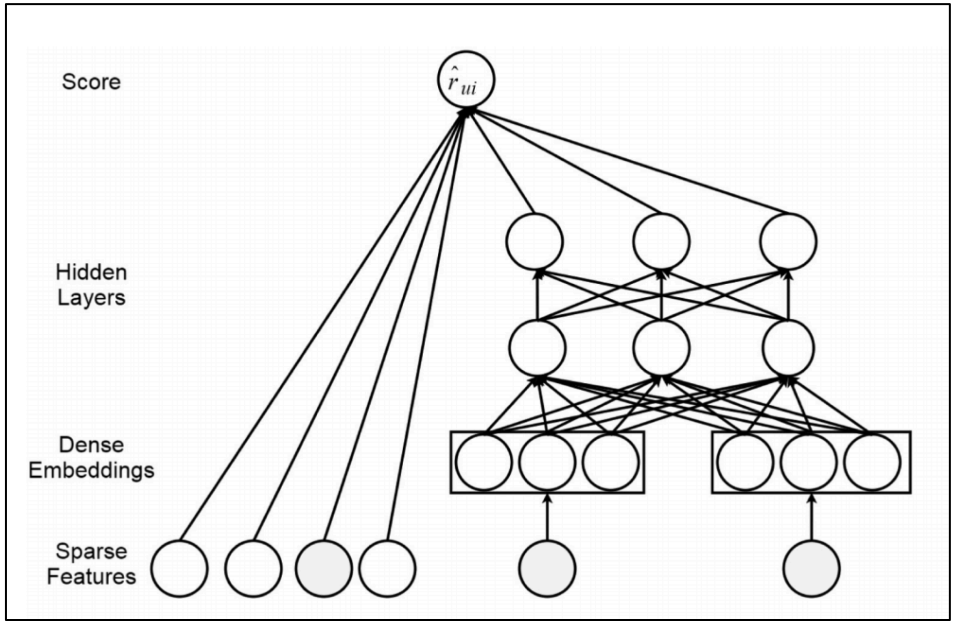
RQ3 Are deeper layers of hidden units helpful for learning from user–item interaction data?

A. 1~4 까지 레이어를 쌓았을 때 높을수록 더 나은 성능을 보였다. 비선형성에서 더 좋은 성능을 보인다

https://arxiv.org/pdf/1708.05031.pdf

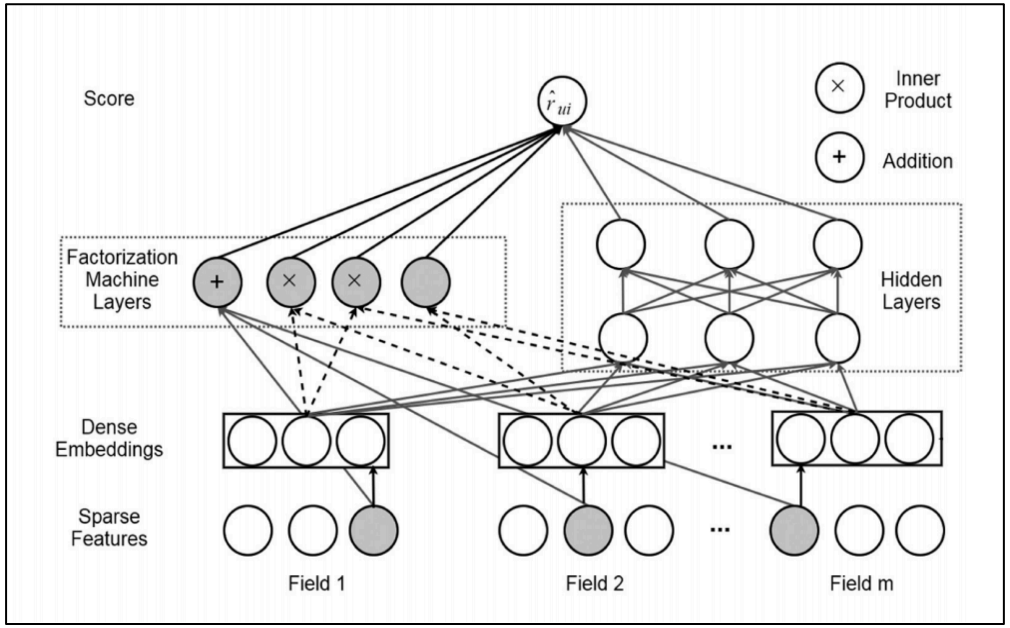
He, Xiangnan, et al. "Neural collaborative filtering." *Proceedings of the 26th international conference on world wide web*. 2017.

1.MLP 기반 모델 : Wide & Deep, DeepFM



Wide & Deep

Feature Engineering O



DeepFM

Feature Engineering X

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

2. Autoencoder 기반 모델

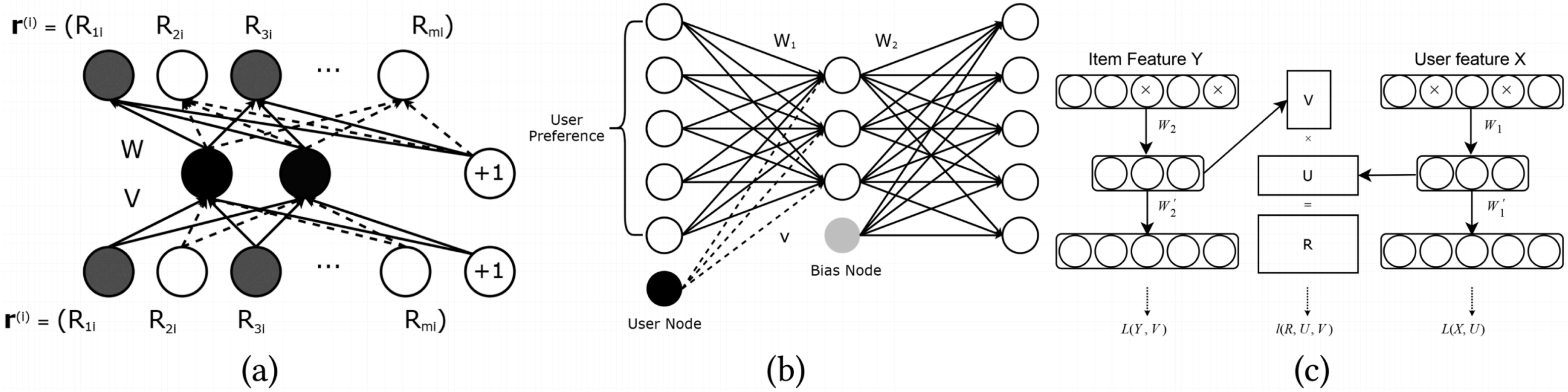
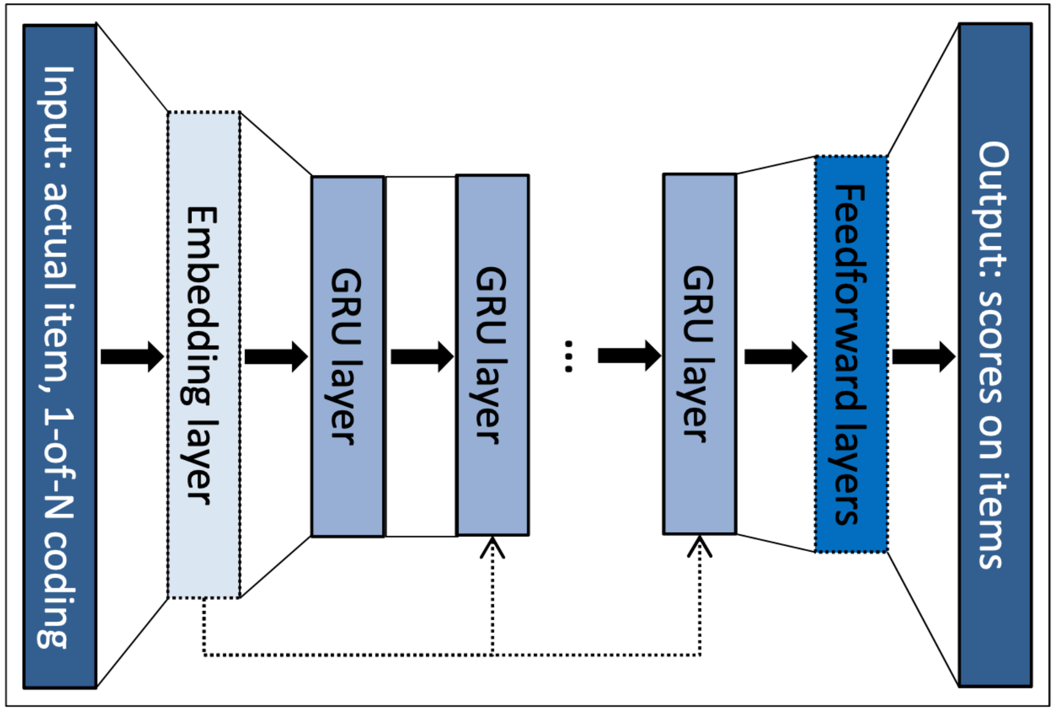


Fig. 4. Illustration of (a) item-based autorec; (b) collaborative denoising autoencoder; (c) deep collaborative filtering framework.

AutoRec -> DAE -> CDAE -> Multi-VAE

3. Session Based Recommendation



1. GRU4REC

2. Attention

Future Research

1. Joint Representation Learning
2. Explainable Recommendation
3. Cross Domain Recommendation
4. Deep Multi-Task Learning
5. Scalability
6. Evaluation

Q & A

들어주셔서 감사합니다.