SESSION-BASED RECOMMENDATIONS WITH RECURRENT NEURAL NETWORKS

Overview

Contents Based

User-based

Item-based

Session-Based?

Session?

네트워크 상에서 사용자(클라이언트)와 컴퓨터(서버) 사이 연결된 상태

Ex) 클라이언트가 원격 컴퓨터에 접속한 상태

Ex) 사용자가 온라인 쇼핑몰에 접속한 상태

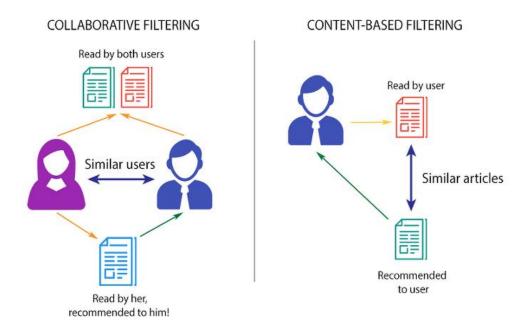
Session-based Recommendation?

사용자 기반 프로필이 굳이 없더라도,

한 세션 안에서 만들어진 시퀀스 만을 가지고

마지막 클릭 이후, 다음 클릭할 아이템은 과연 무엇일까를 예측!

일반적으로..



Factor model (ex. MF) 기준

사용자 로그가 엄청 활발한 플랫폼의 경우 -> OK

But, 그렇지 않은 경우가 많다면 (ex. Cold Start) -> ??













언제 필요한가?

- 유의미한 유저 로그를 수집, 분석하기에는 서비스 크기가 그렇게 크지 않음

휴대폰 케이스 업체 vs 넷플릭스

- 개인정보에 민감한 경우
- 비로그인 사용자가 많은 경우 (눈팀)
- 그때 그때 사용자의 목적에 따른 추천을 하고 싶을 때

장단점

Pros

Cons

사용자가 서비스를 이용하는 목적은 항상 같지 않다.

그저 세션 안에서 벌어진 시퀀스를 분석할 뿐,

실제 사용자의 취향을 반영할 확률이 높다.

사용자의 장기적인 취향은 반영하기 힘들다.

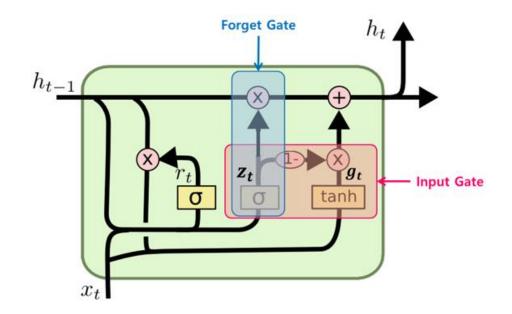
Data

```
Session ID
                Timestamp
                                   Item ID
                                            Label
     3,2014-04-02T13:26:02.515Z,214774687,0
     3,2014-04-02T13:30:12.318Z,214832672,0
     4,2014-04-07T12:09:10.948Z,214836765,0
     4,2014-04-07T12:26:25.416Z,214706482,0
     6,2014-04-06T16:58:20.848Z,214701242,0
     6,2014-04-06T17:02:26.976Z,214826623,0
      7,2014-04-02T06:38:53.104Z,214826835,0
      7,2014-04-02T06:39:05.854Z,214826715,0
     8,2014-04-06T08:49:58.728Z,214838855,0
     8,2014-04-06T08:52:12.647Z,214838855,0
     9,2014-04-06T11:26:24.127Z,214576500,0
     9,2014-04-06T11:28:54.654Z,214576500,0
     9,2014-04-06T11:29:13.479Z,214576500,0
     11,2014-04-03T10:44:35.672Z,214821275,0
     11,2014-04-03T10:45:01.674Z,214821275,0
```

사용자 User Profile을 굳이 필요로 하지 않는다.

Model

Gate Recurrent Unit (GRU)



$$\mathbf{h_t} = (1 - \mathbf{z_t})\mathbf{h_{t-1}} + \mathbf{z_t}\mathbf{\hat{h_t}}$$

where the update gate is given by:

$$\mathbf{z_t} = \sigma(W_z \mathbf{x_t} + U_z \mathbf{h_{t-1}})$$

while the candidate activation function $\hat{\mathbf{h_t}}$ is computed in a similar manner:

$$\hat{\mathbf{h}}_{\mathbf{t}} = \tanh \left(W \mathbf{x}_{\mathbf{t}} + U (\mathbf{r}_{\mathbf{t}} \odot \mathbf{h}_{\mathbf{t-1}}) \right)$$

and finaly the reset gate $\mathbf{r_t}$ is given by:

$$\mathbf{r_t} = \sigma(W_r \mathbf{x_t} + U_r \mathbf{h_{t-1}})$$

Model

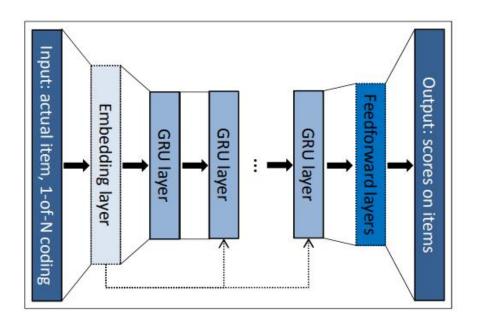


Figure 1: General architecture of the network. Processing of one event of the event stream at once.

Parallel mini-batch

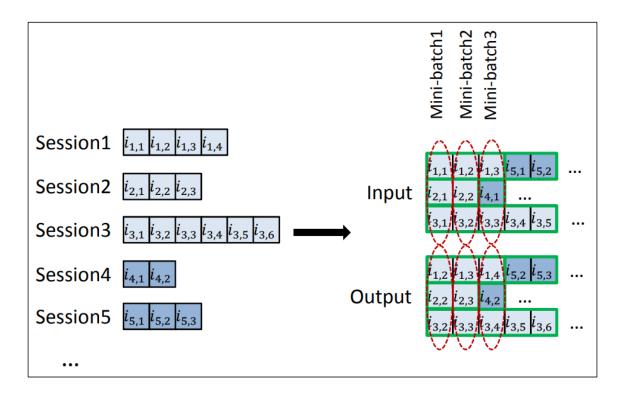
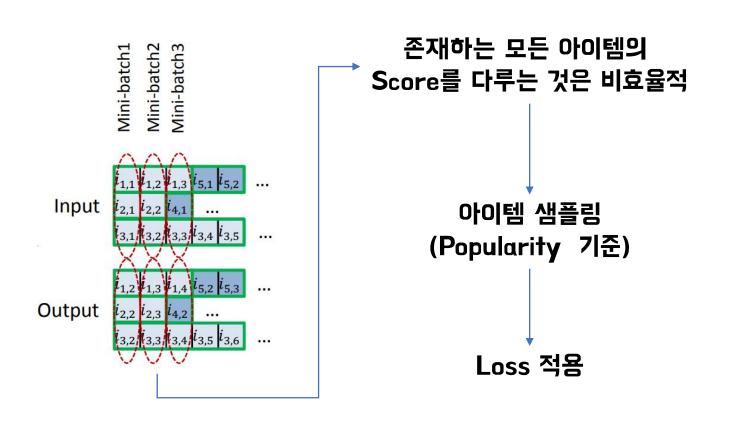


Figure 2: Session-parallel mini-batch creation

보통 세션의 크기는 천차 만별

학습을 효과적으로 하기 위해 한 세션에 데이터가 끝나면 다른 세션을 바로 다음 배치에 넣음

Sampling



Negative Sample

인기 아이템인데 안 봤으면 싫어할 확률이 높다. 다른 세션의 아이템을 Negative Sample로 사용

컴퓨팅 시간 절약

코드 간결화

Ranking Loss

Pairwise ranking loss 가장 잘 동작 함.

BPR
(Bayesian
Personalized
Ranking)

$$L_s = -\frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \log (\sigma (\hat{r}_{s,i} - \hat{r}_{s,j}))$$

Ns : sample size

r : score

i : actual item

s : negative sample

$$L_s = \frac{1}{N_S} \cdot \sum_{j=1}^{N_S} \sigma \left(\hat{r}_{s,j} - \hat{r}_{s,i} \right) + \sigma \left(\hat{r}_{s,j}^2 \right)$$

Regularization term

Experiments - Datasets

RSC (Recsys Challenge 2015)

VIDEO (Youtube-like OTT)

전자상거래 유저의 클릭 기록과 구매기록

유투브 Like Service

기본 전처리 추가

세션이 너무 긴 것은 Bot으로 의심

Experiments - Baseline

Table 1: Recall@20 and MRR@20 using the baseline methods

	Baseline	RSC15		VIDEO	
		Recall@20	MRR@20	Recall@20	MRR@20
	POP	0.0050	0.0012	0.0499	0.0117
	S-POP	0.2672	0.1775	0.1301	0.0863
<u> </u>	Item-KNN	0.5065	0.2048	0.5508	0.3381
	BPR-MF	0.2574	0.0618	0.0692	0.0374

Baseline 채택

- POP: 가장 인기 상품
- S-POP: 현재 세션의 가장 인기 상품
- Item-KNN : 같이 노출된 정도에 따라 아이템에 유사도 부여
- BRR-MF : 주로 사용되는 MF 알고리즘

Result

Table 3: Recall@20 and MRR@20 for different types of a single layer of GRU, compared to the best baseline (item-KNN). Best results per dataset are highlighted.

Loss / #Units	RSC15		VIDEO	
LOSS / #UIIIIS	Recall@20	MRR@20	Recall@20	MRR@20
TOP1 100	0.5853 (+15.55%)	0.2305 (+12.58%)	0.6141 (+11.50%)	0.3511 (+3.84%)
BPR 100	0.6069 (+19.82%)	0.2407 (+17.54%)	0.5999 (+8.92%)	0.3260 (-3.56%)
Cross-entropy 100	0.6074 (+19.91%)	0.2430 (+18.65%)	0.6372 (+15.69%)	0.3720 (+10.04%)
TOP1 1000	0.6206 (+22.53%)	0.2693 (+31.49%)	0.6624 (+20.27%)	0.3891 (+15.08%)
BPR 1000	0.6322 (+24.82%)	0.2467 (+20.47%)	0.6311 (+14.58%)	0.3136 (-7.23%)
Cross-entropy 1000	0.5777 (+14.06%)	0.2153 (+5.16%)	_	_

Result

- GRU가 제일 좋았다. LSTM은 비슷했는데 속도가 더 느렸다.
- GRU 레이어가 하나여도 충분했다.
- 경우에 따라서 GRU 사이즈는 늘리는 것 추천
- 추론속도 빠르고 재학습에 용이

