

# 브런치 연관추천 개선 & MAB 서베이

kakao

Zackie

# 목차

1. MAB survey : NN bandit & CoFineUCB
2. MAB context 재실험
3. Only gc에 limit 걸기
4. 고차원 context (= arm context + world context)
5. LinUCB with disjoint vs non-disjoint
6. LinTS with disjoint model
7. SubUCB
8. 기타

# 1. MAB Survey

## > 1. A Neural Networks Committee for the Contextual Bandit Problem

- 2014 & Orange Labs in France & ICONIP
- NN Bandit
- No simulation

## > 2. Hierarchical Exploration for Accelerating Contextual Bandits

- 2012 & CMU & ICML
- CoFineUCB & SubUCB
- Experiment in Brunch

# 1. MAB Survey : NN Bandit

## > A Neural Networks Committee for the Contextual Bandit Problem

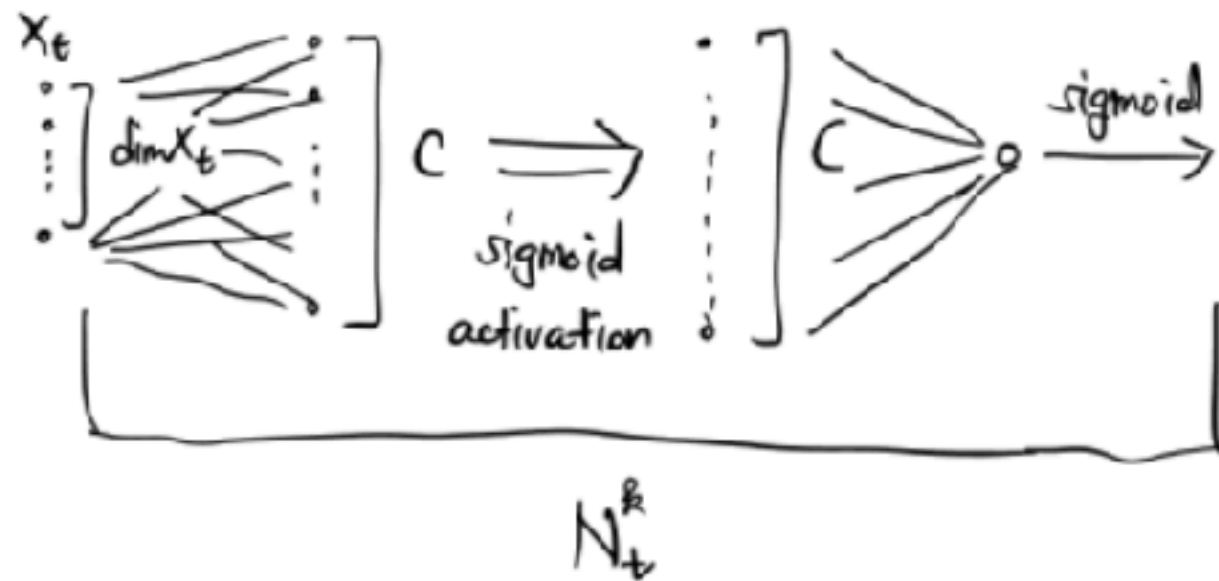
- LinUCB, LinTS의 기본가정 = “Expected reward =  $\theta * \text{context}$ ” (선형 관계)
- 실제로 context와 expected reward가 비선형 관계일 가능성도 있음
- Universal approximator인 NN을 각 arm에 대응시켜 ‘possibly non-linear’인 관계를 학습
- 요약하자면, context가 주어졌을 때 reward를 줄 확률을 output으로 주는 neural net with one hidden layer가 arm마다 주어져있음
- Arm의 원활한 addition, eviction을 위해 거대한 neural net 하나보다는 각 arm에 nn을 대응시켰음

# 1. MAB Survey : NN Bandit

$K$ : # of actions

$C$ : # of neurons of the hidden layer

$N_t^k: X \rightarrow Y$  as  $(x_t) \mapsto$  the output of the NN corresponding to the action  $k$  at round  $t$   
 time  $t$ 의 context



- 여기서 total connection의 개수는  $\dim X_t \cdot C + C = C(\dim X_t + 1)$
- $N := C(\dim X_t + 1)$
- 원형과 같은 Neural Net이 각 arm이라 하나씩 있음.

# 1. MAB Survey : NN Bandit

## > Update Scheme

- **Cost ftn.**: Quadratic Error ftn.
- **Activation ftn.**: Sigmoid ftn.

## > Non-stationarity

- 해당 논문에선 Non-stationary를 다루기 위해 exploration factor  $\gamma$ 를 사용해 적극적으로 exploration을 하게끔 세팅 물론  $\gamma$ 는 constant한 parameter임.

가령, round  $t$ 에서 각 NN의 forward propagation을 통해 arm  $\hat{k}_t$ 가 the highest reward prediction이라고 했을 때

$P_t(k) := (1 - \gamma) \mathbb{1}[k = \hat{k}_t] + \frac{\gamma}{K}$  의 prob. dist.으로 arm을 draw함.

이제 arm의 draw도 stochastic하렸으니 gradient update도 이를 반영해야 하는데, 위의  $\Delta_t$ 에다 단순히  $\frac{\mathbb{1}[\hat{k}_t = k]}{P_t(k)}$ 를 곱해주면 된다. 이렇게 하면 gradient의 기대값이 정확히  $\lambda(\text{cost ftn})'$ 가 되기에...

# 1. MAB Survey : NN Bandit

---

**Algorithm 1:** NeuralBandit1

---

**Data:**  $\gamma \in [0, 0.5]$  et  $\lambda \in ]0, 1]$

**begin**

Initialize  $W_1 \in ]-0.5, 0.5[^{N \times K}$

**for**  $t = 1, 2, \dots, T$  **do**

Context  $x_t$  is revealed

$\hat{k}_t = \arg \max_{k \in [K]} \mathbf{N}_t^k(x_t)$

$\forall k \in [K]$  on a  $\mathbf{P}_t(k) = (1 - \gamma)\mathbf{1}[k = \hat{k}_t] + \frac{\gamma}{K}$

$\tilde{k}_t$  is drawn from  $\mathbf{P}_t$

$\tilde{k}_t$  is predicted and  $y_{t, \tilde{k}_t}$  is revealed

Compute  $\tilde{\Delta}_t$  such as  $\tilde{\Delta}_t^{n, k} = \frac{\lambda \hat{x}_t^{n, k} \delta_t^{n, k} \mathbf{1}[k_t = k]}{\mathbf{P}_t(k)}$

$W_{t+1} = W_t + \tilde{\Delta}_t$

---

# 1. MAB Survey : CoFineUCB

## > Hierarchical Exploration for Accelerating Contextual Bandits

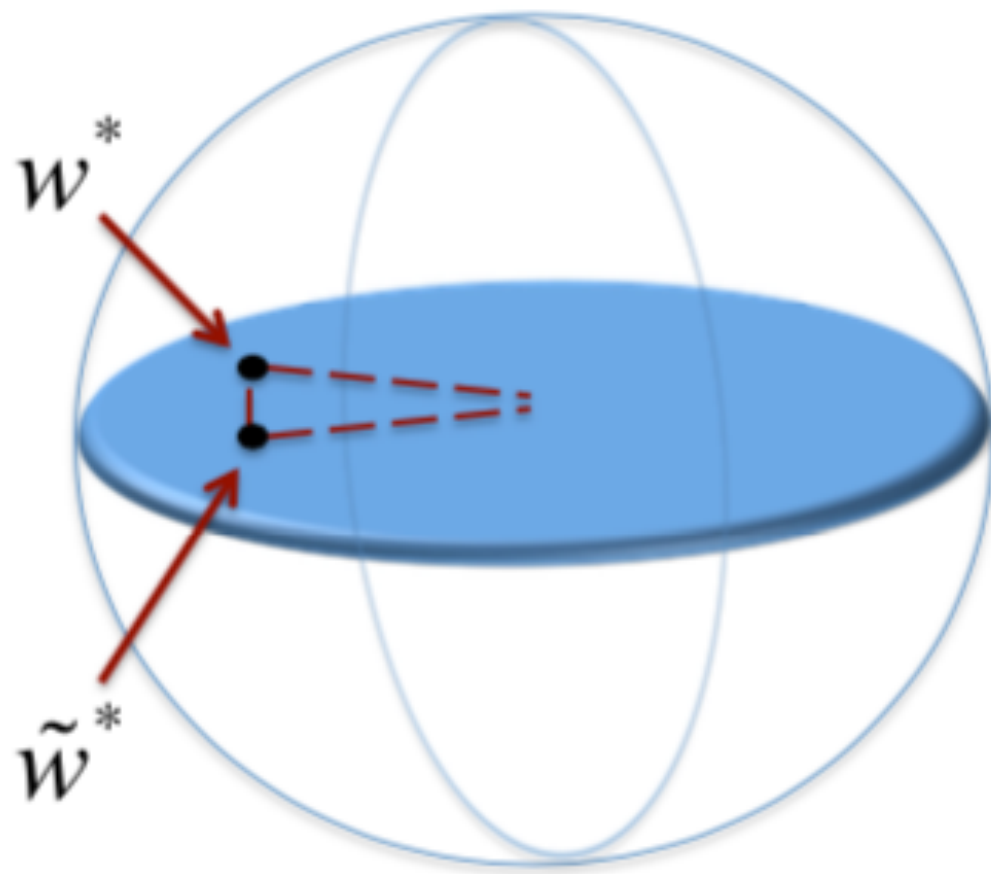
- **Stationary 환경에서...**
  - Lower bound of regret of LinUCB =  $\Omega(\sqrt{Td})$
  - Upper bound =  $O(\sqrt{Td \cdot \ln^3(KT \cdot \ln(T)/\delta)})$
  - $T$  = # of rounds,  $d$  = dim. of context,  $K$  = # of arms
  - Reward를 나타내는 linear payoff의 표현력이 동일하다는 전제조건이 있음
- **Context의 차원을 높일 때 Tradeoff** : regret bound의 상승, 탐색공간 커짐 vs 표현력 증대
  - Context의 차원을 높이면서 parameter  $\theta$ 를 효율적으로 탐색할 방법만 있으면 regret 감소 + 표현력 증대를 기대할 수 있음



# 1. MAB Survey : CoFineUCB

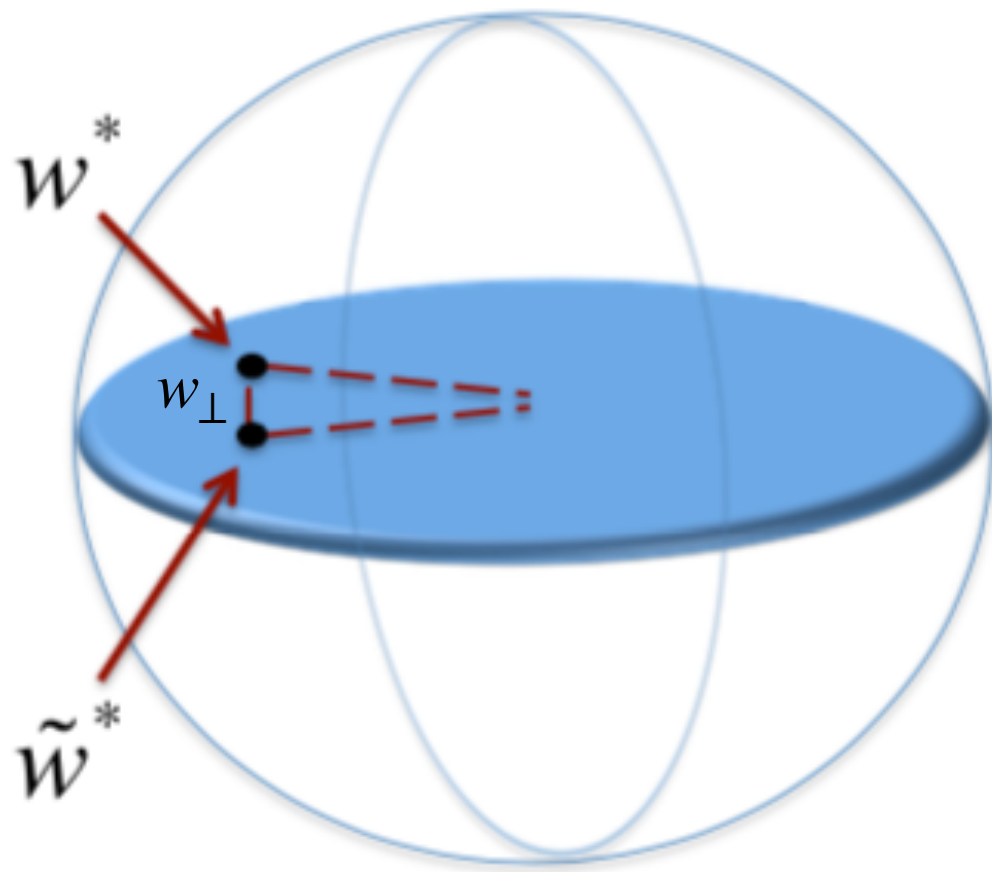
## > Hierarchical Exploration for Accelerating Contextual Bandits

- 해당 논문에선 학습을 통해 탐색이 이뤄지는 공간에 있는 user preference(=  $\theta$ )를 저차원에 embedding해서 효율적인 탐색을 하고자 함



- $w^*$  = 기존 feature space에서의 user preference를 뜻하는  $\theta$
- $\tilde{w}^*$  = 저차원으로 mapping된 user preference
- Prior knowledge를 활용해 탐색에 효율적인 projection을 만들어야 함

# 1. MAB Survey : CoFineUCB



$$w^* = \tilde{w}^* + w_{\perp}$$

- User profile에 해당되는  $w^*$ 을 저차원으로 projection할 때 orthogonal component가 0에 가깝도록 해서  $w^* \cdot x \approx \tilde{w}^* \cdot x$  로 근사
- Ex) User profile을 분석해서 야구, 크리켓을 동시에 좋아하는 경우가 없다는 걸 알게됐을 때, 두 토픽을 projection 했을 때 서로 정반대 방향으로 위치하게끔 subspace를 디자인할 수 있음
- 저차원으로 가서 rough한 탐색을 한 뒤, 다시 고차원으로 올라와 정밀 탐색

# 1. MAB Survey : CoFineUCB

---

**Algorithm 1** CoFineUCB

---

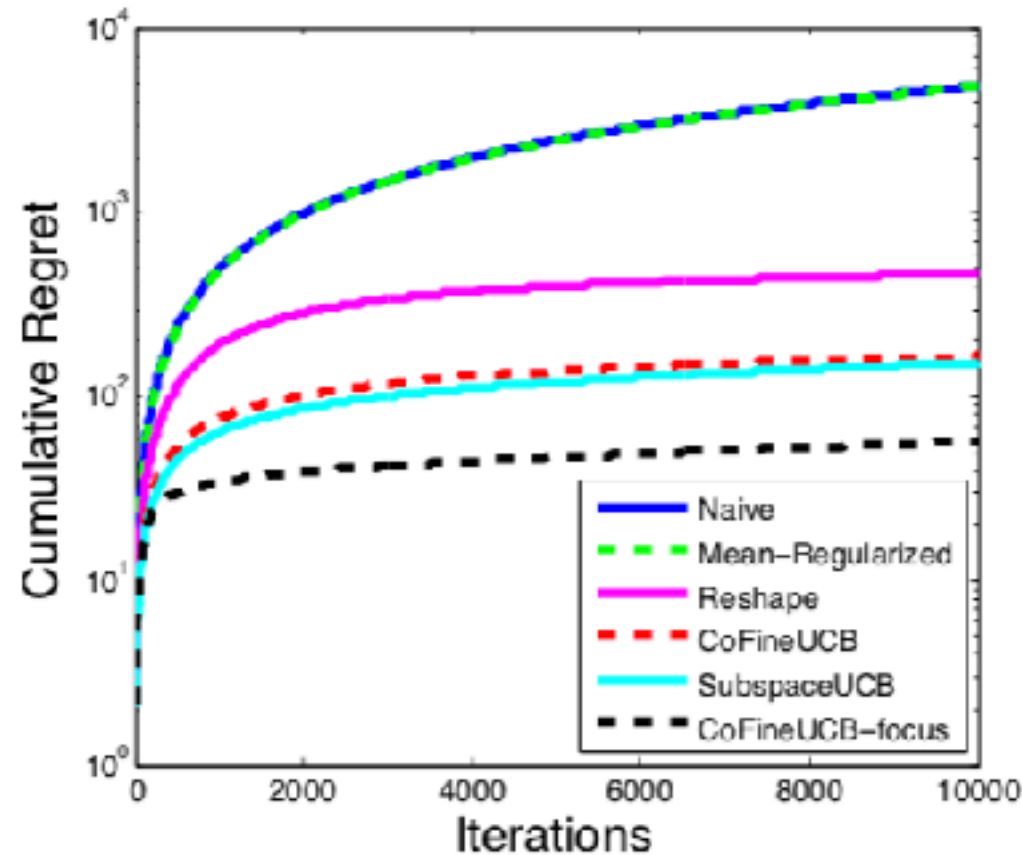
```
1: input:  $\lambda, \tilde{\lambda}, U, c_t(\cdot), \tilde{c}_t(\cdot)$ 
2: for  $t = 1, \dots, T$  do
3:   Define  $X_t \equiv [x_1, x_2, \dots, x_{t-1}]$ 
4:   Define  $\tilde{X}_t \equiv U^\top X_t$ 
5:   Define  $Y_t \equiv [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{t-1}]$ 
6:    $\tilde{M}_t \leftarrow \tilde{\lambda} I_K + \tilde{X}_t \tilde{X}_t^\top$ 
7:    $\tilde{w}_t \leftarrow \tilde{M}_t^{-1} \tilde{X}_t Y_t^\top$  //least squares on coarse level
8:    $M_t \leftarrow \lambda I_D + X_t X_t^\top$ 
9:    $w_t \leftarrow M_t^{-1} (X_t Y_t^\top + \lambda U \tilde{w}_t)$  //least sq on fine level
10:  Define  $\mu_t(x) \equiv w_t^\top x$ 
11:   $x_t \leftarrow \operatorname{argmax}_{x \in \mathcal{X}_t} \mu_t(x) + c_t(x) + \tilde{c}_t(x)$  //play action
    with highest upper confidence bound
12:  Recommend  $x_t$ , observe reward  $\hat{y}_t$ 
13: end for
```

---

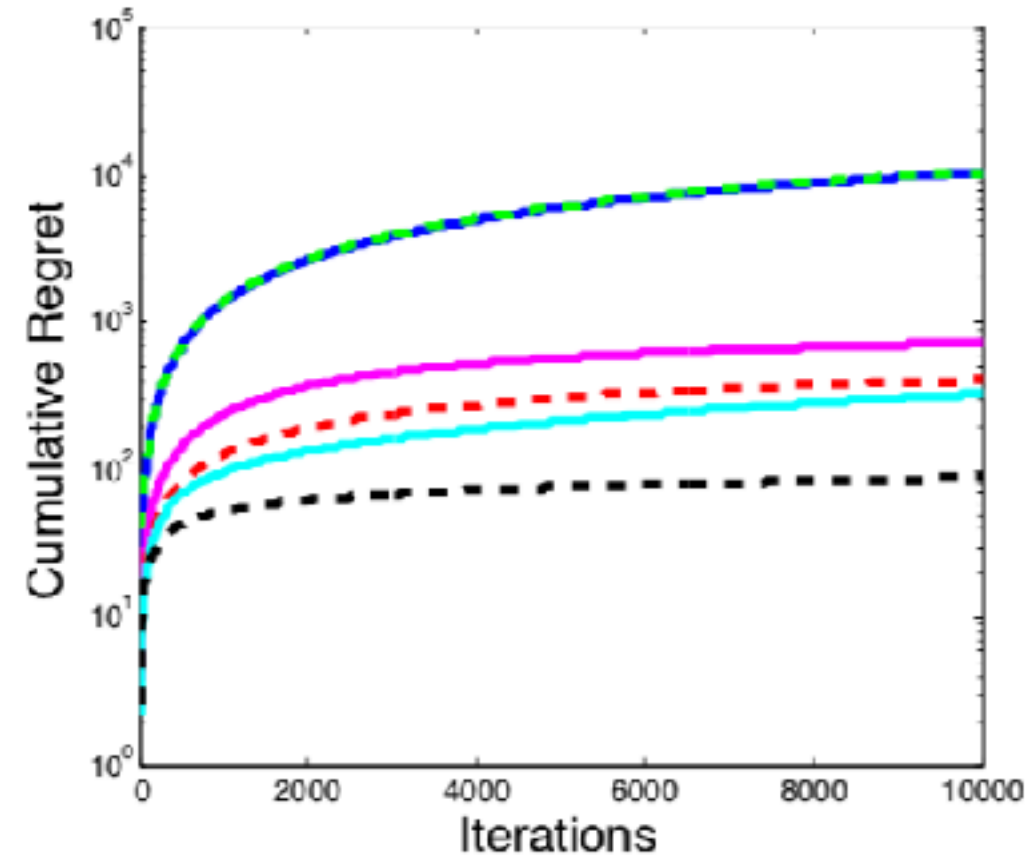
$U$  : projection matrix

$\tilde{X}$  : projection of contexts

# 1. MAB Survey : CoFineUCB



(a) All users simulation ( $L = 1$ )



(b) All users simulation ( $L = 5$ )

- **Naive**: LinUCB
- **SubspaceUCB**: 저차원으로 projection 한 뒤 저차원에서만 탐색하는 LinUCB
- **CoFineUCB-focus** : 특정한 파라미터를 사용했을 때
- $L$  = 뽑는 arms 개수; (b)가 실제 서비스에 가까움

## 2. MAB context 재실험

### > Python 2 vs Python 3

- Python 2 :  $2 / 10 = 0 \rightarrow 2 / 10. = 2 / \text{float}(10) = 0.2$
- Python 3 :  $2 / 10 = 0.2$

### > Error in `get_percentiled_context_features`

```
def get_percentiled_context_features(key):
    meta = metas.get(key, {})
    if not meta:
        return np.zeros(5)

    return np.array(
        [(bisect.bisect_left(
            self.context_stats['view']['percentile'],
            meta['view_count']) + 1) / self.context_stats['view']['num_standards'],
         (bisect.bisect_left(
            self.context_stats['comment']['percentile'],
            meta['comment_count']) + 1) / self.context_stats['comment']['num_standards'],
         (bisect.bisect_left(
            self.context_stats['share']['percentile'],
            meta['share_count']) + 1) / self.context_stats['share']['num_standards'],
         (bisect.bisect_left(
            self.context_stats['index_word']['percentile'],
            meta['index_word_count']) + 1) / self.context_stats['index_word']['num_standards'],
         (bisect.bisect_left(gc_percentile, self.gc.get(key, 0.0)) + 1) / 1000]
    )
```

## 2. MAB context 재실험

### > Python 2 vs Python 3

- Python 2 :  $2 / 10 = 0 \rightarrow 2 / 10. = 2 / \text{float}(10) = 0.2$
- Python 3 :  $2 / 10 = 0.2$

### > Error in `get_percentiled_context_features`

```
def get_percentiled_context_features(key):
    meta = metas.get(key, {})
    if not meta:
        return np.zeros(5)

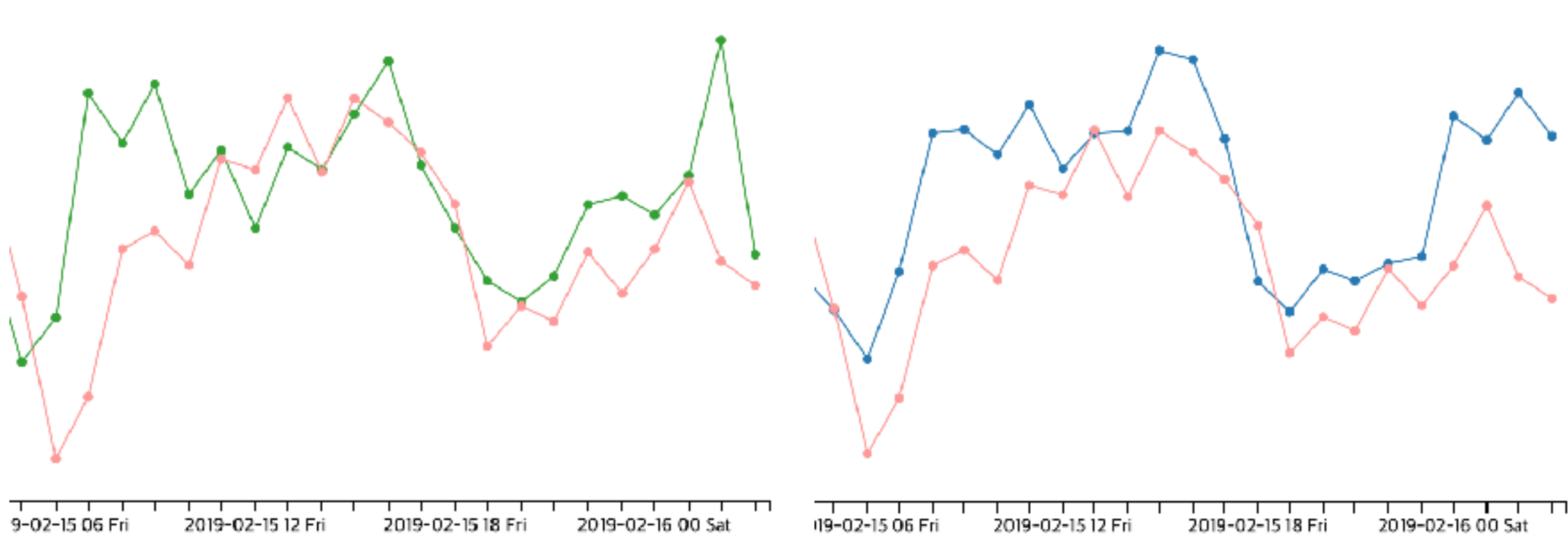
    return np.array([
        (bisect.bisect_left(
            self.context_stats['view']['percentile'],
            meta['view_count']) + 1) / float(self.context_stats['view']['num_standards']),
        (bisect.bisect_left(
            self.context_stats['comment']['percentile'],
            meta['comment_count']) + 1) / float(self.context_stats['comment']['num_standards']),
        (bisect.bisect_left(
            self.context_stats['share']['percentile'],
            meta['share_count']) + 1) / float(self.context_stats['share']['num_standards']),
        (bisect.bisect_left(
            self.context_stats['index_word']['percentile'],
            meta['index_word_count']) + 1) / float(self.context_stats['index_word']['num_standards']),
        (bisect.bisect_left(gc_percentile, self.gc.get(key, 0.0)) + 1) / 1000.]
    ])

ret = {k: get_percentiled_context_features(k) for k, _ in ens}
return ret.iteritems()
```

## 2. MAB context 재실험

> Marv2, Marv4 context 재실험 (under LinUCB)

- **Marv2** = (view, share, comment, #words, gc) + 백분위
- **Marv4** = (view, share, comment, #words, gc, newness) + 백분위 (green)
- **Duzi** = 대조군



## 2. MAB context 재실험

> Marv2, Marv4 context 재실험 (약 24시간)

- Gamma = Marv2 : 상대 ctr 8.65% up
- Alpha = Marv4 : 상대 ctr 4.97% up
- Beta = duzi (대조군)

[Total	]	Impression :	1344698		Click :	173746		CTR :	0.129
[Brunch Sim2 Item Gamma	]	Impression :	435748		Click :	59942		CTR :	0.138
[Brunch Sim2 Item Base	]	Impression :	48901		Click :	2218		CTR :	0.045
[Brunch Sim2 Item Alpha	]	Impression :	429355		Click :	57058		CTR :	0.133
[Brunch Sim2 Item Beta	]	Impression :	430694		Click :	54528		CTR :	0.127

0.0865384004806  
0.0496615110117

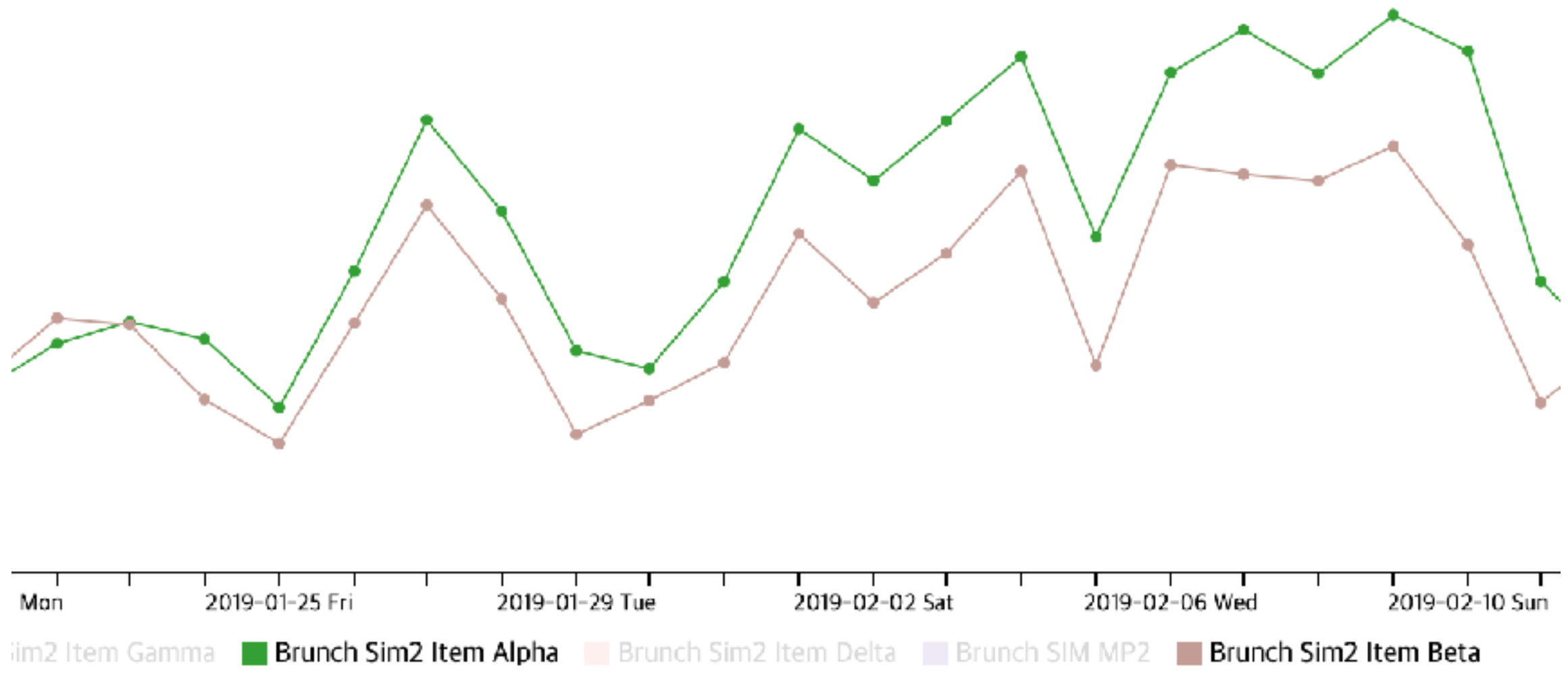


## 2. MAB context 재실험

### > Remark 1

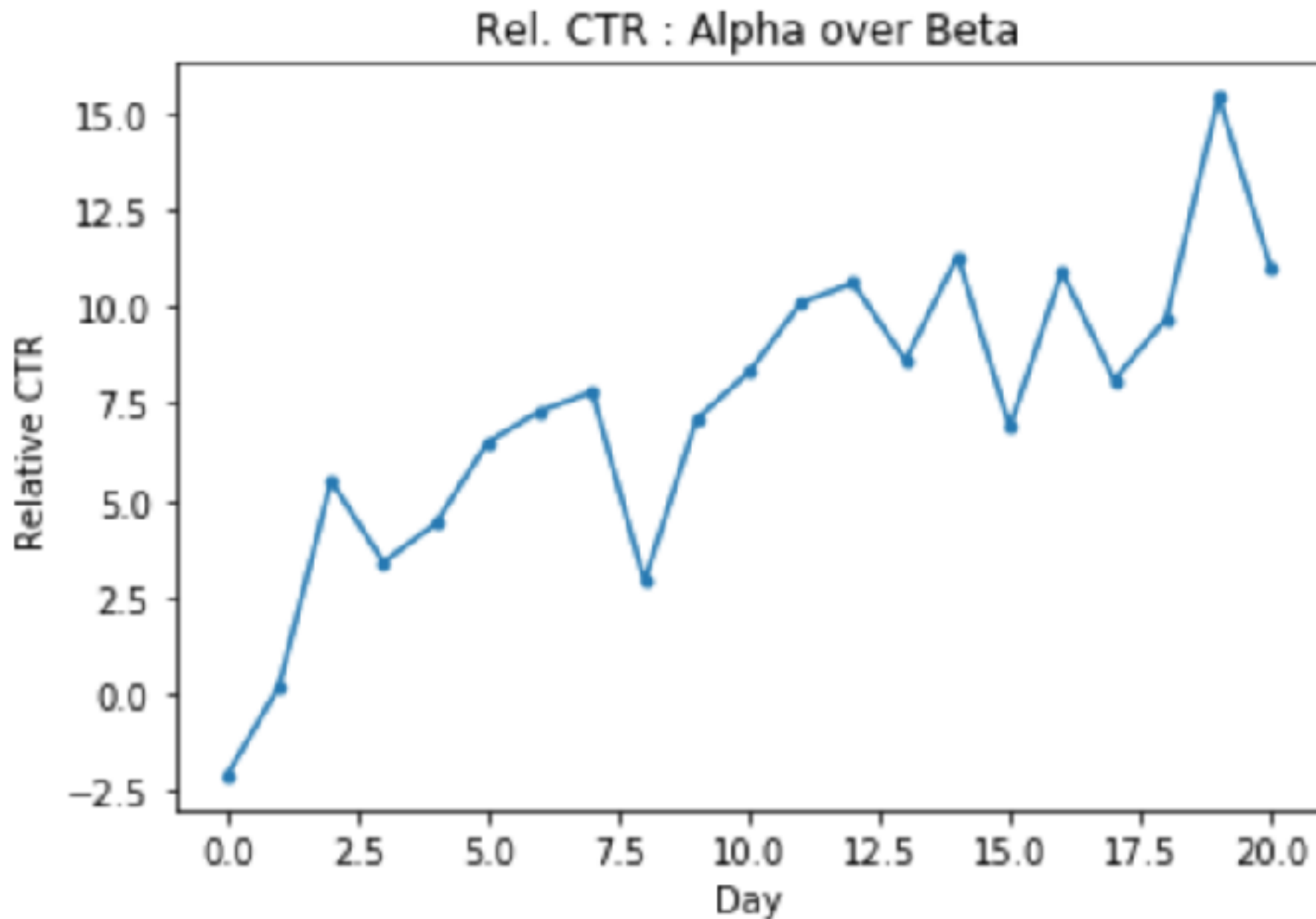
- 예전 버전 marv 4는 사실상 (view, share, comment, gc, newness)인데 v, s, c 값은 거의 0 or 1이었고 상위 0.001%에만 들어야 해당 component를 1로 줬음.
- 해당 context는 1차 프로젝트 발표 당시 그다지 큰 성능 개선을 보여주지 못 했는데...

## 2. MAB context 재실험



- Alpha = 예전 Marv4
- Beta = duzi (대조군)

## 2. MAB context 재실험



- 2일 후부터 성능개선이 보이기 시작하더니 전반적으로 우상향 그래프를 그림.
- 해당 case만 보았을 땐 상대 ctr이 장기적으로 점점 증가할 가능성도 있음.

## 2. MAB context 재실험

### > Remark 2

- New marv2의 성능이 (상대 ctr 8.65%) 제일 높긴 했으나 아무래도 share, comment가 많은 gc가 우선적으로 나오는 경우가 많았음. (예전 버전과 동일한 문제)

<세계 최초 홀 디스플레이, 갤럭시 A9>

```
[hanadmin@aurochs-brunch-similar intern_temp]$ python reco_result.py 6TPQ_45
```

share	comment	words	views	Alpha=====
174	18	3754	852309	아끼면 똥 되는 것 4가지
422	122	2201	708260	나쁜년이 잘 사는 이유 ...
113	50	2241	719028	중국인들이 이해 못하는 한국문화
1357	57	2087	892821	이효리를 통해 바라본 자존감 높은 사람의 특징
143	7	2355	391909	자녀 망치는 부모 행동
107	6	2622	62251	좋은 배우자의 기준 .
18	11	3324	55629	중국인에겐 당연한 일 , 우리에게 글썄 ...?
390	9	2024	299917	남자가 진짜 사랑할 때 11가지 심리와 행동
86	63	1856	761885	생활비를 당장 줄이는 7가지 절약 방법
1892	11	1825	426677	40대 스쿼트 , 푸시업 3개월 했더니 생긴 놀라운

변화

## 2. MAB context 재실험

### > Remark 2

- New marv2의 성능이 (상대 ctr 8.65%) 제일 높긴 했으나 아무래도 share, comment가 많은 gc가 우선적으로 나오는 경우가 많았음. (예전 버전과 동일한 문제)

<애플, 월 10달러 뉴스 앱 출시... 언론사와 반반씩 수입>

```
[hanadmin@aurochs-brunch-similar intern_temp]$ python reco_result.py 234D_2692
```

share	comment	words	views	Alpha=====
55	17	3931	637019	벨기에의 3가지 보물
390	9	2024	299917	남자가 진짜 사랑할 때 11가지 심리와 행동
411	11	1298	335574	러닝머신보다 효과 좋은 체지방 빼는 맨몸 운동 5가지
91	11	1363	390281	세계에서 가장 안전하다는 국가 1위는?
1	20	1971	62549	끊임없는 레깅스 패션에 관한 논쟁
2	2	1571	224756	핵폭탄처럼 충격적인 옷 #비키니를 아십니까?
4	3	2029	13513	40대에 16시간 간헐적 단식으로 이룬 기적
20	5	1955	9162	애플의 내리막길 ?!
0	9	2235	54908	며느리는 백년손님 ... 제가 한번 해봤습니다
174	18	3754	852309	아끼면 똥 되는 것 4가지

### 3. Only gc에 limit 걸기

#### > Use `apply_count_filter` function

- Only gc limit 3 적용
- 실험세팅

alpha : marv2 with no only gc limit

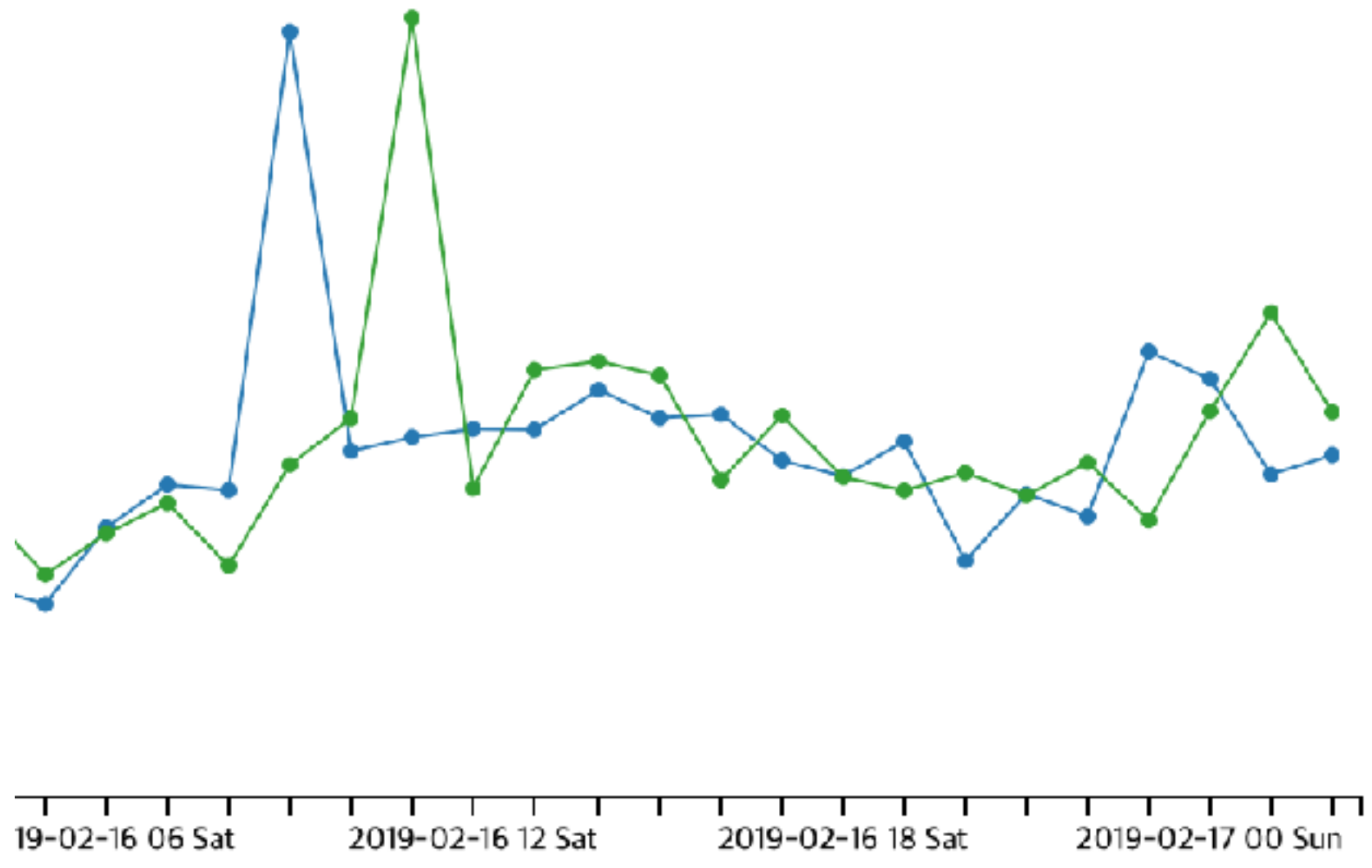
beta : duzi context (대조군)

gamma : marv2 with only gc limit(3) (only gc로만 추천된 아티클은 최대 3개까지만 나올 수 있음)

---

### 3. Only gc에 limit 걸기

> No gc limit (**alpha**) vs gc limit (**gamma**)



### 3. Only gc에 limit 걸기

> No gc limit (alpha, green) vs gc limit (gamma, blue)

[Total	]	Impression :	964066		Click :	127876		CTR :	0.133
[Brunch Sim2 Item Gamma	]	Impression :	300788		Click :	42635		CTR :	0.142
[Brunch Sim2 Item Base	]	Impression :	33834		Click :	1669		CTR :	0.049
[Brunch Sim2 Item Alpha	]	Impression :	301425		Click :	42948		CTR :	0.142
[Brunch Sim2 Item Beta	]	Impression :	328019		Click :	40624		CTR :	0.124

- Alpha CTR : 14.2%
- Gamma CTR : 14.2%



### 3. Only gc에 limit 걸기

#### > 실제 결과

오늘도 그는 뒷담화를 한다 .				
share	comment	words	views	Gamma=====
115	16	1220	717303	정신이 건강한 사람들은 하지 않는 것 - 세 가지 전략
30	9	2245	3973	미국에서도 영어는 늘지 않아 !
41	6	1153	384838	웃으며 대화를 종단시키는 마법의 두 문장
52	23	987	1683394	레전드가 된 의정부고 졸업사진
4	3	3675	842	나도 좋은 사수가 될 수 있을까?(하)
102	8	5198	13307	회사에서 직장 동료와 인정이 붙으면 ...
92	6	1753	1702	즐겁게 일하라는 말의 무례함
0	0	1230	2044	CEO 이미지를 관리해야 하는 이유
176	18	3754	857274	아끼면 똥 되는 것 4가지
0	0	753	0	사장님도 , 알바생도 . 반드시 챙겨야 할 알바서식

- “레전드가 된 의정부고 졸업사진 & 아끼면 똥 되는 것 4가지” 등의 only gc 말고도 직장 생활과 관련된 연관 추천 글들이 초반에 많이 나오고 있음

### 3. Only gc에 limit 걸기

#### > 실제 결과

설경으로 아름다운 겨울 추천 여행지				
share	comment	words	views	Gamma=====
2226	17	1767	1244208	이 대구출신 여성은 어떻게 미국의 3천억대 부자가 됐나
1831	89	1034	1307950	만날수록 괜찮은 남성 특징 8가지
0	0	1680	607	인천 월미도 겨울 산책길
115	16	1220	717303	정신이 건강한 사람들은 하지 않는 것 - 세 가지 전략
0	0	2266	4383	6. 시작이 좋아, 조용한 바닷가에서.
10	1	1055	66648	눈이 오면 더욱 아름다운 국내 여행지 추천
0	2	1775	1850	블레드, 호수라고 다 같은 호수가 아니다
4	0	629	1891	#1 무위자연에서 찾은 '몽돌 해변'
0	0	2562	8	알프스의 만년설 자락을 타고, 스위스 기차 여행!
5	0	1993	1765	부산 맛집 전골의 변신은 무죄

- 타 글에 비해 share, comment가 월등히 높은 3개의 글 + 여행지와 관련들 글이 적절히 잘 섞여서 올라옴.

### 3. Only gc에 limit 걸기

#### > 실제 결과

디자이너의	직급			
share	comment	words	views	Gamma=====
390	9	2024	301727	남자가 진짜 사랑할 때 11가지 심리와 행동
86	63	1856	763038	생활비를 당장 줄이는 7가지 절약 방법
1913	11	1825	428370	40대 스쿼트, 푸시업 3개월 했더니 생긴 놀라운 변화
0	1	841	3232	넷플릭스에는 왜 '빨리 보기' 버튼이 없을까?
0	3	2069	5280	개발자가 부러웠던 기획자
0	0	894	3886	900개 팔로우 취소
0	2	758	18370	직원이 떠나갈 때
176	18	3754	857274	아끼면 똥 되는 것 4가지
0	0	1440	29	1. 강렬했던 첫 번째 팀장님
66	120	3193	610262	노브라로 나가보았다

- 연관성은 떨어져보이나 인기 있는 글들이 추천으로 많이 올라온 편인데, cf\_w2v 및 cf가 only gc를 막아서 그런게 아닐까 함.
- 그래도 연관들 글이 아예 없는게 아닌지라 현재 토픽에 계속 관심이 있으면 “팀장님 & 직원이 떠나갈 때” 등을 읽을 것이고, 흥미가 떨어진다고 느끼면 인기성 글로 다시 빠져나갈 수 있음.

### 3. Only gc에 limit 걸기

#### > 추천 양상

- Only gc의 limit이 없을 땐 초반의 한결같은 인기글 추천에 기존 이용자가 흥미를 잃을 가능성이 있음.
- Only gc 개수의 제한을 걸어서 초반부터 다양한 추천이 나가게 되면 기존 이용자 + 신규 이용자 모두 흥미를 가지고 추천을 들여다 볼 수 있음.
  - 기존 이용자 : 이미 읽었던 인기글 말고도 다른 연관된 글이 나옴
  - 신규 이용자 : 연관글 + 인기글 모두 추천으로 나옴

## 4. 고차원 context

### > Context in contextual bandit

- Arm context = Arm을 특정지어주는 feature 등
  - Arm을 특정지어주는 고정된 cb, image feature
  - Category (binary vector)
  - 연속한 값의 경우 값을 5분위로 나눠 5차원 one-hot vector로 표기 가능
  - 신선도
- World context = User context + 추천이 일어나는 상황에 관한 context
  - 낮, 밤
  - 지역, 언어, 사용자의 나이, 취향

## 4. 고차원 context

### > 연관추천에서?

- Arm context의 적용은 쉽지만, user context (world context) 적용은 어려운 상황
- 따라서 추천을 받는 users의 feature를 source article의 feature로 대체해서 생각함.
  - 해당 source 글을 보는 user들의 preference (= parameter \theta)의 평균이 source 글의 feature와 연관되어 있을거라고 가정
- 기존 120차원의 cb feature를 맨 앞 20차원만 자른 뒤 normalize하여 고정시킨 후 사용
  - source cb (20) + arm cb(20) + 낮/밤(2) + newness(1) = (43) = zackie context

## 4. 고차원 context

> 20차원으로 줄였을 때 most similar

- 맨 위가 source 글

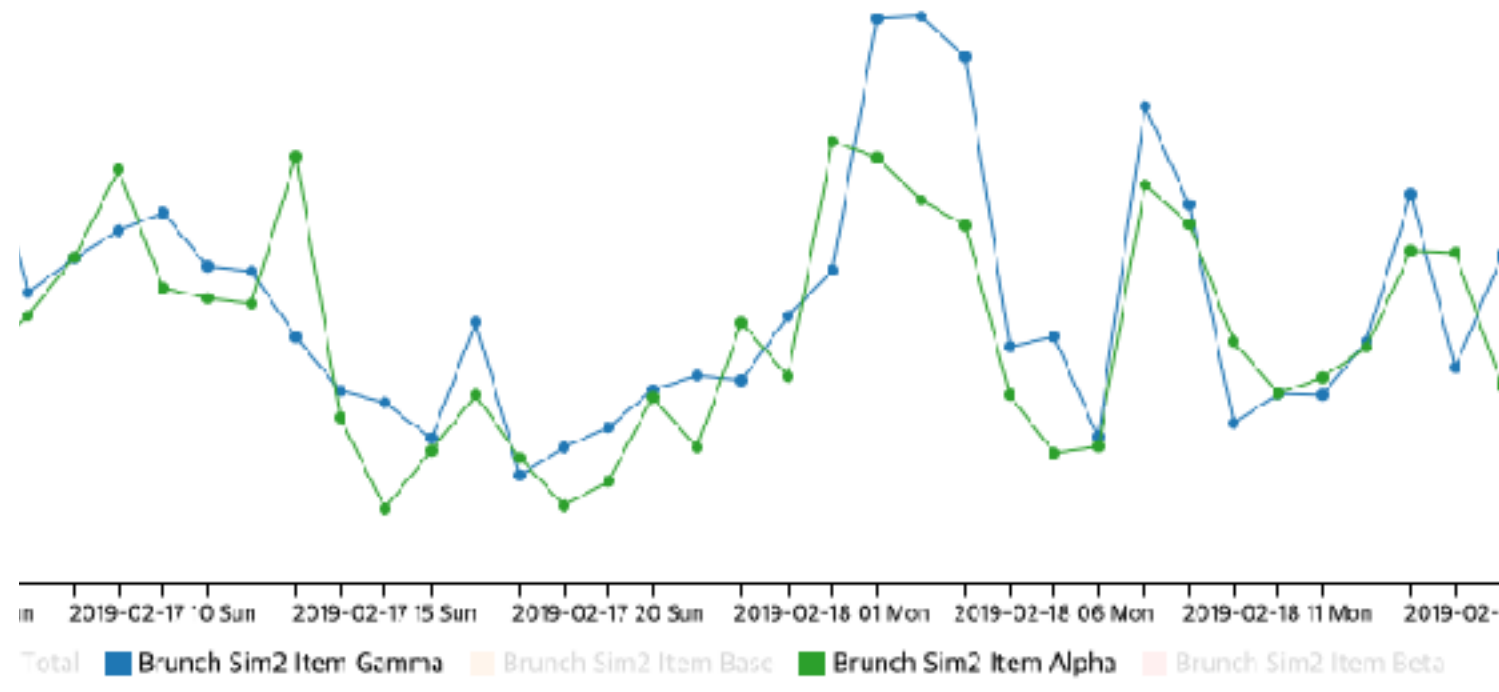
```
>>> similar('39Rl98', 10)
아이폰 XS 맥스 개봉기
아이폰 5se는 과연 경쟁력이 있을까?
아이폰 XS 512G 후기
아이폰 Xs VS 아이폰 Xr 어떻게 다를까?
아이폰 9, 아이폰 XS 맥스 까지! 아이폰 공개 임박
7월 내내 흘러나온 새 아이폰 소문 5가지
아이폰 7 & 아이폰 7 플러스 주요 특징점 살펴보기
갤럭시노트9과 함께! 갤럭시워치, 빅스비 스피커 매그비
커다란 만족을 주는 아이폰 7의 작은 변화
아이폰 7이 궁금해! 키워드로 살펴보는 아이폰7 루머
```

```
>>> similar('2jRLl145', 10)
40대 스쿼트, 푸시업 3개월 했더니 생긴 놀라운 변화
살 빠지는 8가지 좋은 습관
다이어트를 시작할 때 절대 해서는 안되는 일 5가지
20년 헬스경험으로 쓰는 운동에세이 8
살이 안 빠지는 이유와 살 빼는 방법
You are what you eat!
다이어트를 중간에 포기하는 사람들의 안 좋은 습관 5
다이어트 말고 몸 디자인 하자
두둑한 뱃살을 3주만에 완벽하게 없애는 법
건강한 개발자 되기, 30일 일기
```

```
>>> similar('8gBl3', 11)
유튜브 속 K-뷰티와 한류
실제 경험을 바탕으로 한 실무 이야기(뷰티홍보(PR))
남자라면 핑크지! 취향저격 핑크색 공구 Top5
[Why-레페리는 왜 존재하고, 왜 이 일을 하는가.]
[Why-레페리는 왜 존재하고, 왜 이 일을 하는가.]
[유튜브 트렌드 : 언니 오늘은 내가 화장해줄게!
응원하고 싶은 케이블 채널 '라이프타임'
[스타일난다의 뷰티사업으로 본 NCS 전자상거래 마케팅
[브랜드 기획] YOLO 라이프가 만든 남자들의 놀이터
[크리에이터가 그렇게 쉽게 돈을 번다며?
[슈퍼 개인(Influencer)이 생겨나고 있다
```

## 4. 고차원 context

> 결과



- **Alpha** : LinUCB with zackie(43)
- **Gamma** : LinUCB with marv2(5)

[Total	]	Impression :	1183165		Click :	138173		CTR :	0.117
[Brunch Sim2 Item Gamma	]	Impression :	369818		Click :	46649		CTR :	0.126
[Brunch Sim2 Item Base	]	Impression :	42230		Click :	2095		CTR :	0.050
[Brunch Sim2 Item Alpha	]	Impression :	369390		Click :	45639		CTR :	0.124



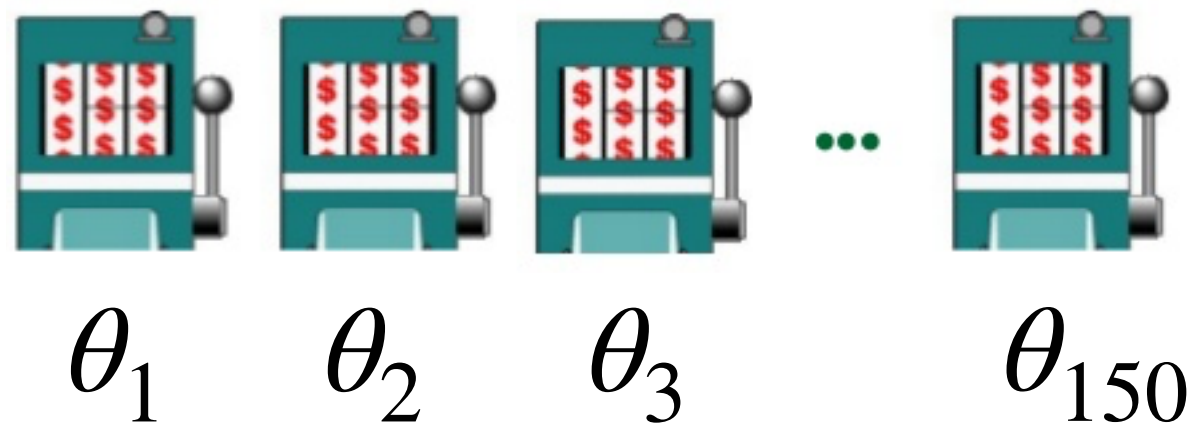
# 5. LinUCB with disjoint vs non-disjoint

## > Disjoint vs Non-disjoint

- **Recall** : LinUCB의 중요한 가정 = “Expected reward =  $\theta * \text{context}$ ”
  - $\theta$  : unknown parameter to be learned
- 기존에 사용하던 LinUCB는 disjoint model인데, 뜻하는 바는 추천 pool에 있는 arm들끼리 parameter를 독립적으로 가지고 있고 서로 영향을 주지 않음 (뒤 슬라이드 참고)
- 좀 더 흔히 소개되고 있는 LinUCB with non-disjoint model은 arm끼리 parameter를 공유해서, parameter 자체가 bandit에 귀속되는 형태임.
  - 해석: Bandit에 귀속돼서 트레이닝 되는 parameter  $\theta$ 가 결과적으로 source article을 보는 user들의 “평균”의 **preference**를 나타낸다고 생각할 수 있음.
- Disjoint model vs Non-disjoint model의 성능 차이?

# 5. LinUCB with disjoint vs non-disjoint

## > Disjoint vs Non-disjoint



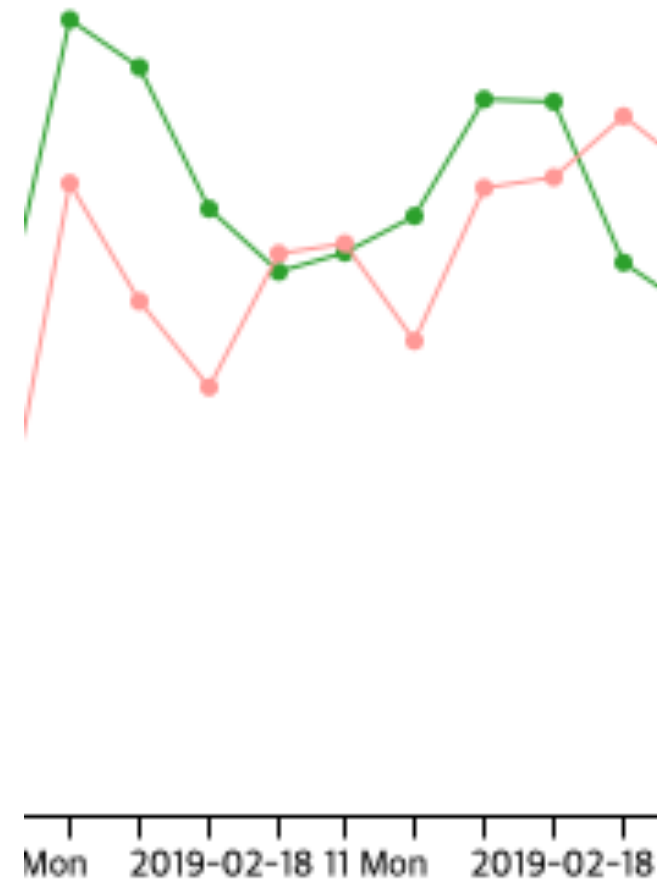
- 저장할 파라미터들 개수 많음
- 트레이닝 데이터가 부족할 수 있음
- Non-stationarity를 handle

- Training이 좀 더 용이
- 저장할 파라미터 개수 적음
- Arm pool이 바뀌는 환경 적응 어려움

# 5. LinUCB with disjoint vs non-disjoint

## > 실험결과

- **Green** : LinUCB with disjoint model
- **Red** : LinUCB with non-disjoint model
- Context : zackie context로 동일
- 디스크 마운트 작업 전까지 9시간 비교
- **Green** : 12.7%
- **Red** : 11.9%



## 6. LinTS with disjoint model

### > 과거 Duzi's issue history

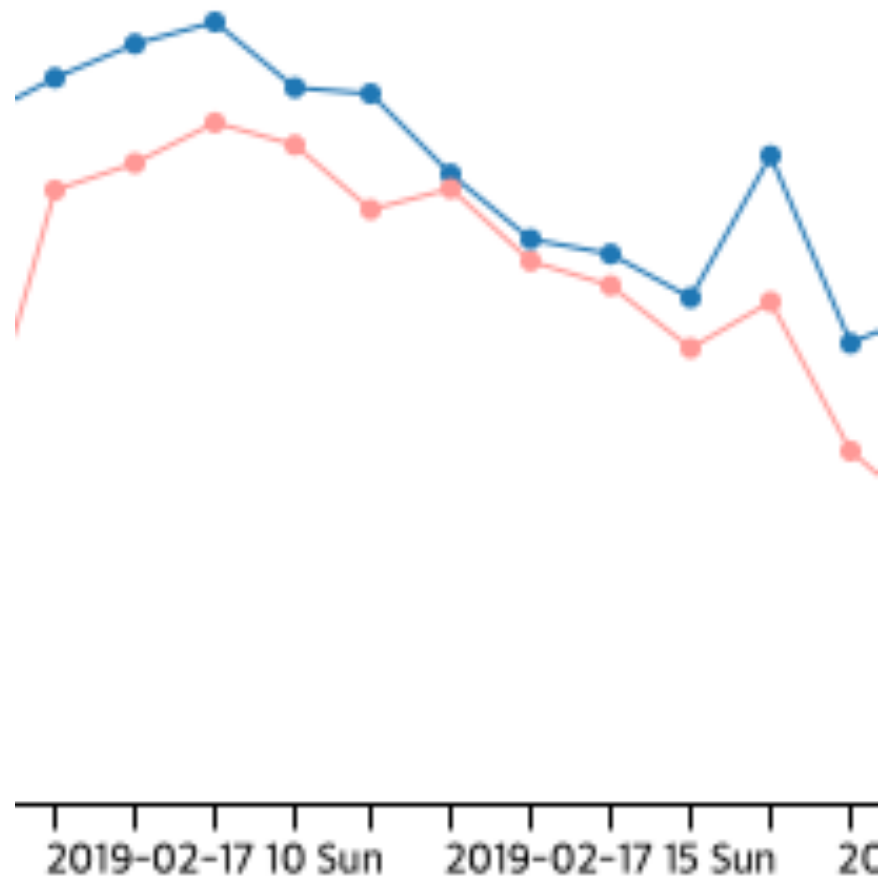


- **Recall** : LinTS = Contextual Thompson Sampling with Linear payoff
  - LinUCB와 동일하게  $\text{expected reward} = \theta * \text{context}$ 라는 가정으로 시작
  - $\theta$ 를 multivariate normal distribution  $N(u, \text{std})$ 에서 draw하는데  $u, \text{std}$ 를 학습시켜 나감
- duzi가 구현했던 방식은 bandit 하나당 random draw를 하는 distribution이 하나가 있는 형태인 LinTS with disjoint model
  - 5번 실험을 토대로 disjoint model 적용

## 6. LinTS with disjoint model

### > 실험결과

- **Beta** : LinTS with disjoint model
- **Gamma** : LinUCB with disjoint model (대조군)
- Context : marv2 동일



- **Beta** : 12.1%
- **Gamma** : 12.9%
- 당시 duzi의 LinTS with non-disjoint의 비교 모델은 지금의 Gamma보다 성능이 떨어지는 duzi context의 LinUCB였는데 비슷하게도 따라 붙는 걸 보니 disjoint가 non-disjoint보단 좀 더 효과적이란 생각이 더욱 더 듭

# 7. SubUCB

## > SubspaceUCB

### - Idea

- 앞서 LinUCB with zackie context로 학습되었던  $\theta$ 들을 이용해서 projection matrix를 만들고 이것을 prior knowledge & 저차원 mapping tool로 사용
- CoFineUCB는 저차원에서 rough하게 찾고, 고차원에서 다시 한 번 자세히 찾는 알고리즘
- SubspaceUCB는 저차원으로 projection해서 subspace에서 LinUCB
  - 앞서 학습된 prior knowledge(projection matrix)로 저차원에서의 효율적 탐색을 기대할 수 있음
- Disjoint model도 만들어서 실험해 봄

# 7. SubUCB

## > SubspaceUCB

### - Projection Matrix?

- LinUCB with zackie context에서 구한 parameter \theta를 crawling함.
- 충분한 impression이 쌓인 후 얻어진 \theta에 대해서만 SVD를 진행함.
- **D = 43 to K = 5**  $\rightarrow \mathbf{W} \in \mathbb{R}^{D \times K}$

---

### Algorithm 2 LearnU: learning projection matrix

---

- 1: **input:**  $W \in \mathbb{R}^{D \times N}$ ,  $K \in \{1, \dots, D\}$
  - 2:  $(A, \Sigma, B) \leftarrow SVD(W)$
  - 3:  $U_0 \leftarrow A_{1:K}$  //top  $K$  singular vectors
  - 4: Solve for  $\Omega$  via (16) using  $U_0$  and  $W$
  - 5: **return:**  $U_0 \Omega^{1/2}$
- 

root 빠진듯?  $\rightarrow \Omega = \frac{K}{\text{trace} \sqrt{\tilde{W}_0 \tilde{W}_0^\top}} \sqrt{\tilde{W}_0 \tilde{W}_0^\top}, \quad (16)$

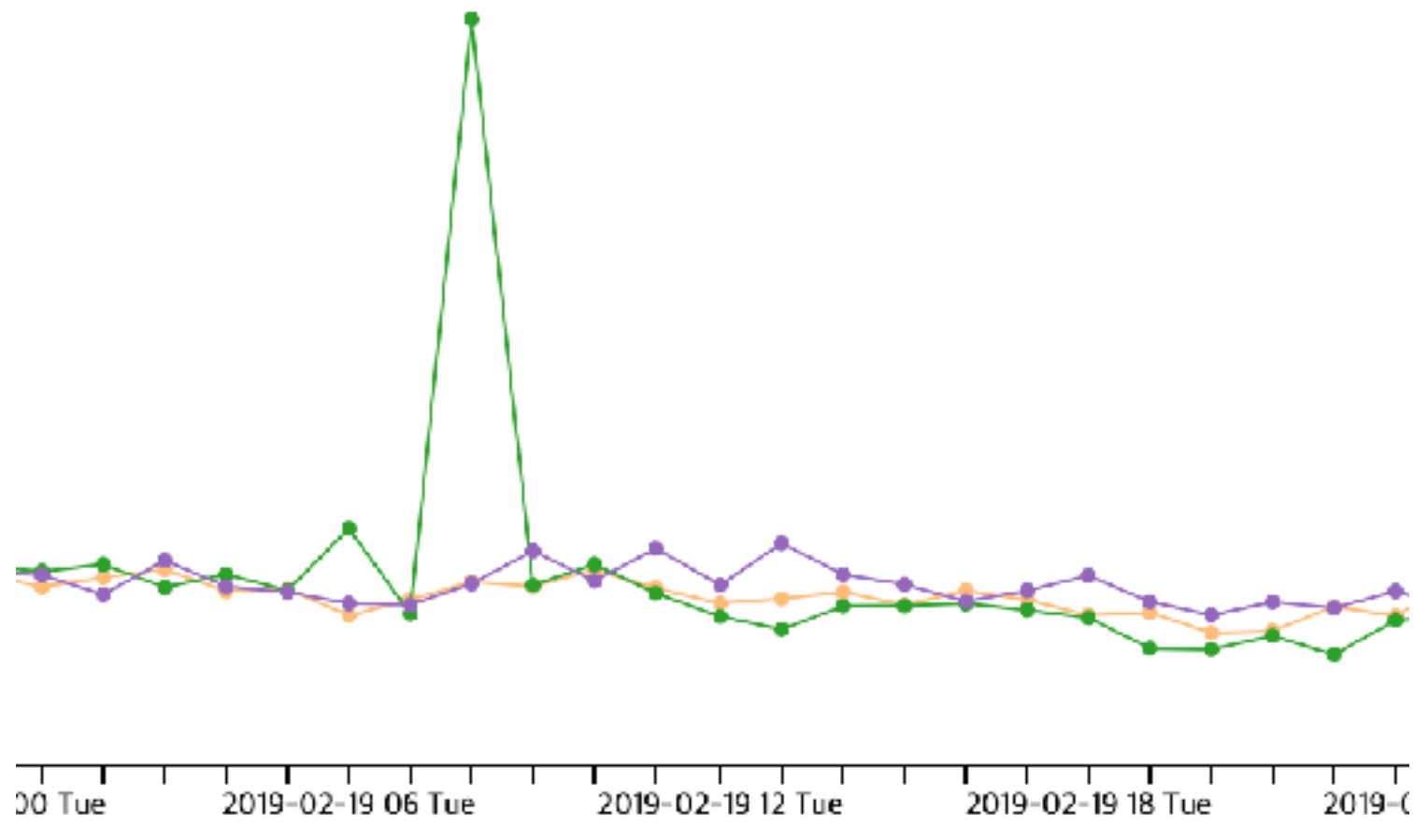
where  $\tilde{W}_0 \equiv (U_0^\top U_0)^{-1} U_0^\top W = U_0^\top W$ . See Appendix

# 7. SubUCB

## > SubspaceUCB

### - Setting

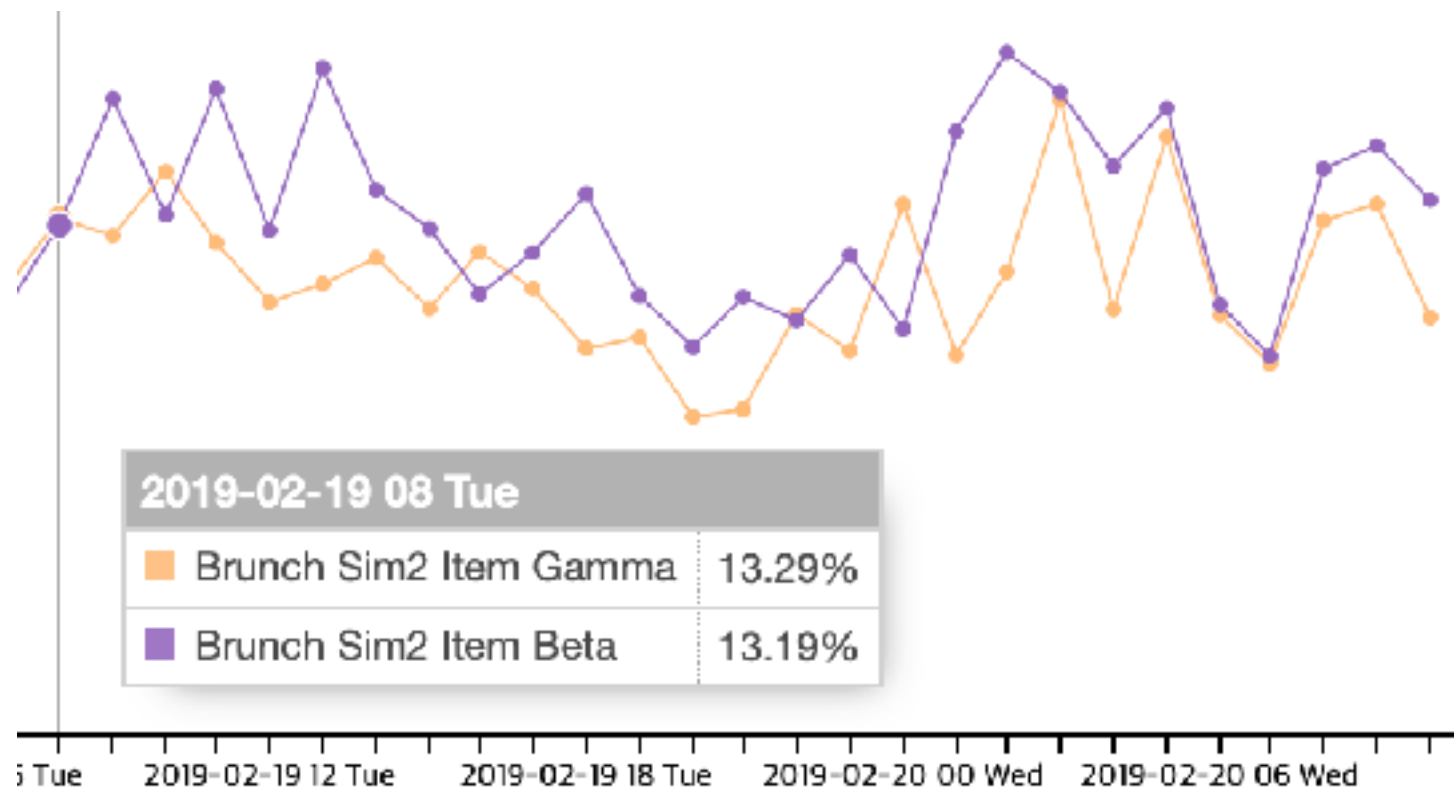
- **Alpha** : SubUCB w/ non-disjoint model (zackie)
- **Beta** : SubUCB w/ disjoint model (zackie)
- **Gamma** : LinUCB w/ disjoint model (marv2); 대조군





# 7. SubUCB

> SubspaceUCB



**Beta** : SubUCB w/ disjoint model (zackie)

**Gamma** : LinUCB w/ disjoint model (marv2); 대조군

# 7. SubUCB

## > SubspaceUCB

[Total	]	Impression :	676595		Click :	80158		CTR :	0.118
[Brunch Sim2 Item Base	]	Impression :	24527		Click :	997		CTR :	0.041
[Brunch Sim2 Item Gamma	]	Impression :	219103		Click :	26376		CTR :	0.120
[Brunch Sim2 Item Alpha	]	Impression :	212682		Click :	23980		CTR :	0.113
[Brunch SIM MP2	]	Impression :	606		Click :	53		CTR :	0.087
[Brunch Sim2 Item Beta	]	Impression :	219677		Click :	28752		CTR :	0.131

**Alpha** : SubUCB with non-disjoint model (zackie)

**Beta** : SubUCB w/ disjoint model (zackie)

**Gamma** : LinUCB w/ disjoint model (marv2); 대조군

Beta over Gamma: 8.72% Up

## 8. 기타

### > Memory 문제 해결?

- 20차원으로 줄인 후 고정시켜놓은 cb feature (120MB) 때문에 추가적인 메모리 필요
  - 서버에서 Reco.py 3개가 돌고 있는데, 각 Reco에서 로드했더니  $120 \times 3$ 가 더 필요하고..
  - 경고 메시지 + 1시간마다 업데이트 되는 120차원 cb feature를 로드할 때 마다 터짐
- mab cache를 위한 SSDB에다 넣어서 메모리 부담을 덜었음
- 디스크 용량이 부족하자 넉넉하게 500G를 더 물려주었기에...
- 가장 좋은 건 현재 8G 램을 늘리는 게...

# Thanks

kakao

Zackie