

Review : Online Models for Content Optimization

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, Nitin Motgi, Seung-Taek Park,
Raghu Ramakrishnan, Scott Roy, Joe Zachariah

August 4, 2022

Abstract

- 현재 주요 야후 포털에서 사용하는 사용자에게 제공할 기사를 선택하는 새로운 콘텐츠 게시 시스템을 설명한다.
- 편집자들이 정기적으로 표시할 기사를 선택하는 수동 접근 방식에 비해 사용자 클릭 수를 크게 증가시킨다.
- 다이나믹한 콘텐츠 풀, 짧은 기사 수명, 고정되지 않은 클릭률, 매우 높은 트래픽 볼륨 등을 해결해야 한다.
- 클릭률이 낮은 기사는 신속하게 폐기해야 한다.
- 온라인 모델을 통해 실시간으로 기사의 성능 추적을 기반으로 한다.
- Online Model에 대한 분석은 많은 연구방법을 제시한다.

1 Introduction

- 사용자가 웹 포털을 방문할 때, 콘텐츠를 전달하는 효과적인 알고리즘 접근방식을 개발하는 것은 많은 관심을 받지 못하는 것은 근본적인 문제이다.
- 검색엔진과 온라인광고는 자동화된 알고리즘을 이용하여 결과를 보여준다.
- 포털(야후, 네이버 등)은 일반적으로 수동적으로 프로그래밍되어 콘텐츠의 관련성, 화제성, 개인 선호도를 평가하기 어렵기 때문에 사용자를 만족시키기 어렵다.
- 콘텐츠의 수동 방식은 높은 품질을 보장한다. 하지만 기사 수와 프로그래밍하고자하는 사이트의 페이지 수가 증가하면 확장하는데 있어, 비용이 많이 든다.
- 데이터 기반의 머신러닝 알고리즘을 사용하면, 규모의 문제를 해결할 수 있다.
- 위 시스템은 야후 포털에서 현재 사용중이며, 하루에 수억 명의 사용자 방문 서비스를 제공하고있다.
- 기사 순위를 매기는 머신러닝은 "Offline data"(과거에 수집된 데이터)를 사용한 모델이다.
- "Offline data"를 사용하여 실시간데이터를 지속적으로 업데이트하여, 수명이 짧은 기사들에 순위를 매기는데 있어 활용성이 좋다.
- 본 논문은 고전적인 방식이 아닌 이러한 머신러닝적 모델링을 제안하고자 한다.

2 Problem Description

- 야후 포털의 모듈은 메일, 날씨, 뉴스 등 다양한 콘텐츠링크를 제공한다.
- 모듈은 F1, F2, F3, F4라는 4개의 슬롯이 있다.
- F1에 표시된 기사는 다른 슬롯에 표시된 기사들보다 더많은 클릭수를 기록한다.
- 편집자가 새로운 기사를 주기적으로 프로그래밍된 시스템에 푸시하여, 오래된 기사를 대체한다.
- 프로그래밍된 시스템으로 편집된 사이트는 특정 시간에 방문한 모든 사용자에게 동일하게 보여진다.(개인설정에는 영향을 받지않는다.)
- 전체 클릭률(CTR=시간당 총 클릭 수/총 조회수)를 최대화하여 기사를 선택하는 것에 중점을 둔다.
- 가장 중요한 F1 슬롯에서 얻은 클릭 피드백만을 학습에 사용한다.

2.1 System Chllenges

- Highly dynamic system characteristics : 기사 수명은 짧으며(6~8시간), 각 기사들은 표시되는 시간, 슬롯의 위치에 따라 각각 다른 CTR을 가지며, 클릭과 페이지 view에 기반한 모델을 통해 사용자 피드백에 대한 빠른 반응을 가진 성능의 모델이 중요하다.
- Scalability: 포털은 초당 수천 개의 페이지 view를 수신하고 하루에 수억 건의 사용자 방문 서비스를 제공한다. 데이터 수집, 모델 학습 및 기사 채점(모델 사용)은 대기시간이 필요하다. 이벤트(User의 클릭, 페이지 view)는 front-end web서버에서 수집되어 데이터 수집 클러스터로 지속적으로 전송되며, front-end web서버는 정기적으로 업데이트된 모델을 가져오고 모델을 기반으로 콘텐츠를 제공한다.

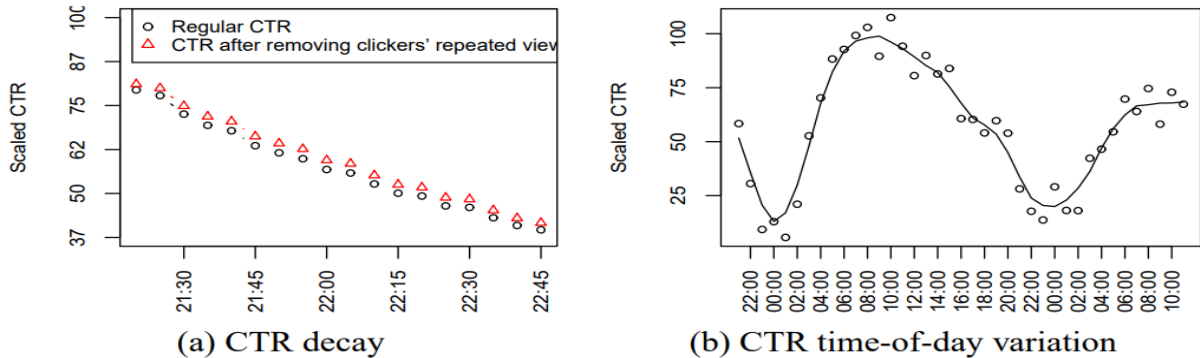


Figure 1: (a):모델 F1의 bucket에 연속적으로 표시될때 해당 기사의 CTR 감소를 보여줌. (b):랜덤 bucket에 기사의 CTR을 표시.

2.2 Machine Learning Challenges

- serving scheme은 사용자에게 모듈의 위치에 표시할 기사를 결정하는 알고리즘이다.
- 알고리즘이 생기기전에는 편집자들이 직접 선택하였는데, 이것을 editorial serving scheme라고 부른다.
- 사용자 모집단의 랜덤 샘플을 bucket이라 한다.
- editorial serving scheme를 사용하는 동안 수집된 과거 데이터를 기반으로 오프라인 모델을 구축하는 시도했다.
- 사용자 기능에는 사용자 방문 패턴에 따른 연령, 성별, 지리적 위치, 유추 관심사가 포함됐다.
- 기사의 경우 URL, 기사 카테고리(예: 스포츠, 엔터테인먼트) 및 제목 키워드를 기반으로 했으나, 성능이 좋지 않았다.
- 이유는 동일한 기사에 대한 CTR의 변동성, 시간에 따른 기사의 CTR의 변화, 비랜덤화된 serving scheme에서 수집된 데이터들을 조정하기 어렵기 때문이다.
- 또한, 비슷한 특징을 기사들에 대한 CTR의 높은 변동을 보여주는 것을 확인하였다(예: 스포츠 기사, 엔터테인먼트 기사).
- Online model 사용하여 주어진 사용자에게 최고의 기사를 선정해주는 방식을 통해 더 나은 성능을 달성했다.
- editorial serving scheme을 Online model로 바꾸어 기사를 선정하는 데에 있어 비용이 증가가 필요하다.

- 해결해야 할 과제-

- Non-stationary CTRs: 기사의 CTR은 사용되는 serving scheme(특히 F1 노출량)에 크게 의존하며 시간이 지남에 따라 극적인 변동이 존재할 수 있다. 따라서, 프로세스 정상성을 가정하는 머닝러닝에는 적용할 수 없다.
- Strong Serving Bias: serving scheme에서 생성된 데이터를 사용하여 구축된 모델은 scheme에 의해 편향된다. 주어진 data가 아닌 다른 영역의 data는 학습되지 못한다. 편향되지 않은 기사를 얻기 위해 이러한 요소들을 조정해야한다.
- Interaction with the editorial team: 포털에 수작업으로 기사를 배치한 편집자들과 상호작용이 필요하다. automated serving schemes이 편집자들의 경험을 활용할 수 있게 만드는 것이 큰 과제다. 편집자들이 제공하는 제약 조건을 통해 pool을 제어하여 알고리즘을 생성해야한다.

3 Experimental Setup and Framework

Experimental Setup: live traffic의 일부로부터 mutually exclusive buckets을 생성하고 serving scheme 중 하나를 사용하여 각 bucket에서 traffic을 처리했다. editorial serving scheme을 실행하는 제어 버킷을 만들었다. 또한 random bucket이라는 별도의 bucket을 만들어 기사를 round-robin fashion으로 제공한다.

-Framework-

- Batch Learning: 모델에 대한 업데이트를 5분간격을 수행하게 세팅한다.
- Business Logic, Editorial overrides: 알고리즘에 의해 기사 위치가 세팅되어도, 편집자가 제약조건을 추가해야 할 경우가 발생할 수 있다.
- Online models to track CTR: 다양한 기사의 CTR을 추적하기 위해 Online model을 구축한다. 기사들의 CTR을 추적하여, serving bucket에서 탐색하고 위치를 변화시켜 점수를 매긴다.
- Explore/Exploit Strategy: random bucket은 확률 p로 무작위 탐색을 하여, 확률이 큰 Online model의 점수에 따라 순위를 매겨 기사들을 제공한다. 또한 비 random bucket에서의 편향 해소(날짜별 CRT패턴)를 하는데 도움이 된다.

4 Online Models

Online model을 이용한 콘텐츠 최적화에 사용하는 3가지 모델을 소개한다.

4.1 Estimated Most Popular: EMP

모델은 시간의 흐름에 따라 F1 위치에서 기사당 CTR의 log-odds를 추적한다. t^{th} 는 bucket에서 첫번째 기사가 표시된 후 t번째 간격이다. c_t , n_t 는 시각 t에서의 클릭수와 view 수를 나타낸다. 로지스틱 변환에 의해 $y_t = \log(c_t + 0.5) - \log(n_t - c_t + 0.5)$, 분산 $w_t = (c_t + 0.5)^{-1} + (n_t - c_t + 0.5)^{-1}$ 로 정의할 수 있다. random bucket에서 5분 간격으로 F1위치에서 300~400개의 관찰값을 얻을 수 있다. 따라서 위의 변환은 사용자 segments가 필요없는 EMP나 SS가 적합하다.

$$y_t = o_t + \mu_t + \epsilon_t \sim N(0, V_{w_t}) \quad (1)$$

$$\mu_t = \mu_{t-1} + \beta_t + \delta\mu_t \sim N(0, \sigma_{\mu_t}^2) \quad (2)$$

$$\beta_t = \beta_{t-1} + \delta\beta_t \sim N(0, \sigma_{\beta_t}^2) \quad (3)$$

- o_t : offline model(예: 하루 중 시간 보정치)에서 얻은 상수 offset.
- μ_t : 시간 t에서의 y_t 의 평균
- β_t : t-1에서 t까지의 시차에서 interpretation of incremental decay
- $\delta\beta_t$: 확률적 요소
- $\delta\mu_t$ 와 $\delta\beta_t$ 는 상관관계가 없는 것으로 가정한다.

Model parameters는 random bucket의 기사에 대한 t=1의 값을 관찰하나, 실제 tracking은 t=2에서 시작된다. the evolutions $\delta\mu_t$ 와 $\delta\beta_t$ 를 0으로 가정한 후, 가중 최소 제곱을 통해 y_t 값에 적합되는 선형 추세인 $\mu_t = \mu_0 + t\beta_0$ 를 통해 state parameters $\theta_t = (\mu_t, \beta_t)$ 를 구할 수 있다. 0이 아닌 evolution의 추가를 통해 선형의 시간의 흐름을 추적하는데 도움을 준다. noise variance(V_{w_t})에 대한 state evolution variance $\sigma_{\mu_t}^2$ 와 $\sigma_{\beta_t}^2$ 의 값은 model에 발생하는 일시적 smoothing의 양을 결정한다. model fitting은 Kalman filter을 활용하여 시행한다[7].

4.2 Saturated Segmented Model: SS

이 모델은 사용자 기능을 통합하기 위해 EMP를 일반화하여 사용자 covariates(공변량)은 disjoint subsets(segments)을 만드는 데 사용한다. local EMP 모델은 각 사용자 segments 항목의 성능을 추적하는데 사용된다. 소수 사용자 segment의 경우, F1에 있는 특정 항목에 대해 사용자 segment마다 별도의 EMP 모델을 적합시킨다. 기사 수명 초기에는 segment의 수가 증가함에 따라 데이터 희소성으로 인해 작은 segment에서 높은 분산 추정치가 발생하는데, 해결하기 위해 베이지안 계층적 모델을 사용한다.

$(a_{it}, Q_{it}), i = 1, \dots, k$ 가 예측된 평균이고 시간 t에 따라 K개의 다른 사용자 segment에서 F1에서 항목 점수의 분산이면 다음과 같이 새로운 점수를 도출한다.

$$\tilde{a}_{it} = \frac{\tau}{\tau + Q_{it}} a_{it} + \frac{Q_{it}}{\tau + Q_{it}} \bar{a}_t$$

\bar{a}_t 는 항목에 대한 EMP model의 점수이다. 가장 인기있는 것으로 알려진 "shrinkage(수축)"의 양인 상수 τ 는 메타 분석에 사용되는 DerSimonian and Laird estimator [10]로 구할수 있다.

4.3 Online Logistic Regression: OLR

SS는 많은 수의 기능이 작동할 때 lower order interactions(교호작용)을 통합할 유연성이 없다. 예를 들어, 사용자의 연령, 성별 및 지리적 위치에 따라 SS 모델은 세 변수의 가능한 모든 조합을 고려한다면, 2-요인 상호 작용(연령-성별, 연령-위치, 성별-위)에 기반한 additive model이 나은 성능을 제공할 수 있다. 그러한 모델을 구축하기 위해 Online Logistic Regression: OLR을 사용한다. OLR은 positive 또는 negative로 라벨링된 event에 대한 parameter를 업데이트한다. EMP와 SS에서처럼 데이터를 경험적으로 변환하여 추가하는 대신, 각 event에 대해 베르누이 가설성을 가정하고 log-odds를 Taylor series expansion을 통해 quadratic approximation(2차 근사)를 통해 선형성을 유지한다[1].

5 Experiments

3가지 온라인 모델(EMP, SS, OLR)을 기반으로 한 serving schemes를 현재의editorial programming approach 방식(ED라고 함)과 비교하는 실험 결과를 제시한다. 한달동안 포털의 실시간 트래픽에서 동시에 4가지 모델을 bucket 테스트를 한 결과, Online Model이 ED를 크게 능가한다는 것을 보여준다. offline 분석을 통해 개인화(OLR의 사용자 기능이나 SS의 segment에 기반한 사용자 기능)가 개선되지 않은 이유를 확인했는데, 주로 SS와 OLR이 EMP보다 발전된 모델이지만 충분히 다양한 기사를 활용하지 못했기 때문이다.

-Bucket Testing Methodology: 평가한 모델은 (1) EMP, (2) 연령*성별 segment가 있는 SS, (3)특징적인 용어가 있는 OLR(Aritcle+Position+Age*ContentCategory+Gender*ContentCategory),(4)ED이다. 1달동안 bucket에서 traffic처리를 하였다. 4개의 bucket은 동시에 실시되었다.

-Online Model Comparison: ED보다 Online model이 CTR을 증가시켰다. 또한 CTR의 증가가 더 많은 사용자가 기사를 클릭하도록 유도하였다고 볼 수 있다.

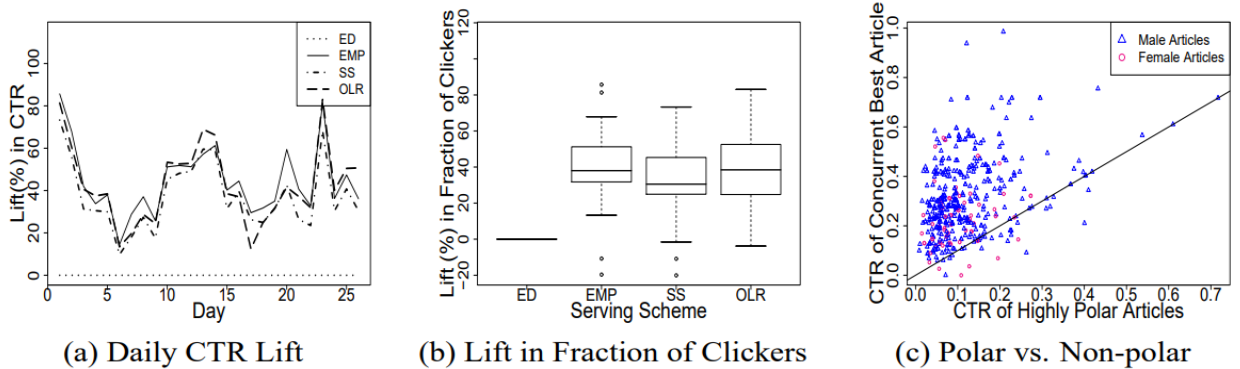


Figure 2: 실험 결과: (a)와 (b)는 bucket 테스트 결과를 보여준다. (c) x축은 segment에 있는 a polar article의 CTR 이고, y축은 동일한 segment에 있는 global best article의 CTR이다

3가지 온라인 모델(EMP, SS, OLR) 모두 ED보다 월등히 우수하며, lift in CRT는 30% ~ 60% 범위에 있다. 사용자 segment를 사용하는 SS와 OLR와 EMP와 크게 다르지 않다. 이는 SS와 OLR가 EMP에 비해 상대적으로 더 나은 예측 가능성을 가지고 있지만, 선택된 사용자 segment에 대한 개인화가 EMP에 대해 소급적용되지 않는다는 것을 의미한다. Figure(2)의 b는 클릭하는 사용자의 비율을 보여준다. online model에 의해 달성된 lift가 ED보다 더 많은 클릭수를 이끄는 것을 보여준다.

-Analysis of Personalization: 소수 사용자의 segment에 더 친화적인 기사(a polar article: 소수segment의

CTR이 전체 CTR의 두 배이상인 기사)가 있다. 모든 사용자에게 인기 있는 기사(non polar article)를 판별하는 것이 더 중요하다.

-Retrospective Evaluation Metrics vs. Bucket Tests: ED에서 얻은 데이터를 가지고 얻은 예측 지표를 가지고 serving scheme(EMP, SS ,OLR)를 평가한다. 평가를 통해 얻은 성능차이가 실시간 traffic에서 얻은 성능차이와 상관성이 없다는 것을 발견하였다. 이는 무작위 bucket 데이터, 편향을 수정하는 serving scheme를 비교하기엔 적절하지 못하여, 신속한 bucket test 인프라의 필요성을 보여준다.

6 Related Work

- google news 개인화 작업: 협업 필터링을 사용하여 전세계 기사를 추천하기 전 사전작업./불량 기사를 걸러내는 것이 목적이며, 메일과 같은 긴 시간에 걸쳐 업데이트되는 정확한 기능을 바탕으로 한 오프라인 모델을 구축하기 위해 노력중이다.
- 널리 사용되는 UCB1체계[8] 역시 저조한 성과를 보인다.
- 콘텐츠 집계 사이트인 digg.com의 최근 연구에서는 Wu 등의 스토리 인기에 대한 모델을 구축/ 편향된 소급 데이터에 기초하고 있어, 실시간 트래픽에서 수행되는 테스트의 결과를 얻을 수 없다.

7 Discussion

- 사용자에게 제시할 기사를 선택하는 Online Model에 대해 설명하였다. 기사의 인기를 신속하게 예측하고 추적해야 한다.
- 오프라인 기능 기반 모델을 통해 효과적으로 생성된 Online model을 통해 특정 사용자(또는 사용자 segment)를 위한 최상의 기사를 추천하고(예:CTR), 분석되지 않은 탐색을 수행함으로써 효율 극대화(예:CTR)것이 필요하다.
- 소규모 콘텐츠 Pool은 충분히 대응가능하지만, 대규모 Pool을 다루는 것에 대해 상당한 진전이 필요해보인다.