AI in Kakao

더욱 똑똑해진 AI 광고 알고리듬

카카오의 광고랭킹 알고리듬

'알파벳(Alphabet Inc., Google의 모기업'[1])은 어떤 회사인가?'라는 질문에 많은 사람들은 검색 서비스 회사나 안드로이드 OS를 만드는 회사 정도로 답한다. 이 글을 읽는 분들이라면 텐서플로우(TensorFlow)를 만든 회사 또는 알파고(AlphaGo)를 만든 딥마인드(DeepMind)의 모회사 정도로 답할지도 모른다. 같은 질문을 페이스북(Facebook Inc.)에 대해 한다면, 소셜미디어(SNS), 인스타그램(Instagram) 혹은 왓츠앱(WhatsApp) 서비스를 제공하는 회사라는 답변이 가장 많을 것이다. 한국을 대표하는 기업들 중 하나인 삼성전자는 스마트폰을 팔아서 돈을 벌고, 현대자동차는 자동차를 팔아서 돈을 버는 반면, 구글과 페이스북은 제품이나 서비스를 파는 것이 아닌 광고를 통해 대부분의 매출을 올리고 있다. 현대자동차가 자동차 회사라면, 구글과 페이스북은 광고 회사인 셈이다. 구글은 광고를 위해 검색 서비스와 안드로이드 OS를 만들고, 페이스북도 광고를 위해 타임라인과 인스타그램 등의 서비스를 제공한다. 광고의 정의와 범위에 따라 달라지겠지만 우리가 알고 있는 대부분의 인터넷 기업들을 광고 회사로 봐야 한다. 광고 비즈니스를 알면 인터넷 기업의 진면모를 제대로 볼 수 있지만, 아쉽게도 많은 이들이 – 심지어 IT 회사의 종사자들마저도 – 인터넷 기업들이 어떻게 돈을 벌고 있는지 자세히 알지 못한다.

글 | 정부환 ben.hur@kakaocorp.com

그냥 논문 읽고 연구하고 논문 쓰기를 좋아하던 학생이 생계를 위해서 검색 데이터를 모델링하고, 쇼핑과 뉴스 등의 컨텐츠를 추천하는 알고리듬을 개발하고, 광고 플랫폼에서 정밀한 타게팅을 고민하다가 지금은 광고랭킹을 개선하면서 10년의 세월을 보냈습니다. 그러는 사이에 빅데이터의 흐름을 경험하고 최근에는 딥러닝을 필두로 한 AI 기술의 폭발에 기술적 고민보다는 철학적 고뇌에 더 깊이 빠집니다.

이 글은 여러분이 기대하는 AI, 특히 딥러닝(deep learning)에 관해서는 다루지 않는다. 이게 무슨 반트렌드적인 발상인가라고 의문을 표할 수도 있겠으나, 회사의 근간을 이루는 기술일수록 더 보수적으로 적용되기 마련이다. 여전히 은행의 레거시 시스템(legacy system)은 코볼(COBOL)로 짜여졌다는 설이 있듯이, 기업의 중요한 자산일수록 더 검증된 기술을 사용하는 경향이 있다. 광고 시스템에 사용되는 알고리듬은 정확도도 중요하지만 짧은 시간 안에 안정적으로 서비스하는 것이 더 중요하다. 그렇기에 부분 기술로는 딥러닝을 적용한 케이스나 논문들이 최근 소개되고 있지만, 광고 시스템 전체를 딥러닝으로 대체하는 데는 여전히 시간이 필요하다. 2006년 10월에 넷플릭스(Netflix)는 추천 정확도(root mean square error, RMSE)를 10% 높이는 알고리듬을 개발한 팀에게 100만 달러를 상금으로 주겠다는 넷플릭스 프라이즈(Netflix Prize)를 진행했고, 실제 목표를 달성한 알고리듬이 2009년에 개발됐다. 하지만, 그 알고리듬을 바로 제품 단계(production level)에서 넷플릭스 추천 엔진에 탑재하지 못했다[2]. 이렇듯 딥러닝 기술도 광고 시스템에 적용하기 위해서는 확장(scale up) 과정이 필요하다.

광고를 단순히 기업의 매출 수단으로만 생각한다면 광고의 본질을 볼 수 없다. 흔히 서비스와 광고를 배타적인 관계로 보는 경향이 있다. 서비스에 광고가 들어오는 순간 서비스의 정체성을 상실한다고 생각한다. 그렇기에 광고가 서비스의 사용자 경험을 해치지 않게 자연스레 녹아들고 때론 더 유용한 정보를 제공하는 매개가 되도록 노력한다. 재미있고 창의적인 광고 소재를 만드는 것에 더해서, 그 광고가 노출될 서비스 또는 매체의 본질을 파악하고 그걸 소비하는 사용자들을 제대로 이해하는 것이 필요하다. 이 글에서 설명할 광고랭킹은 사용자, 매체/서비스, 광고 (소재와 광고주), 그리고 맥락을 연결하는 것이다.

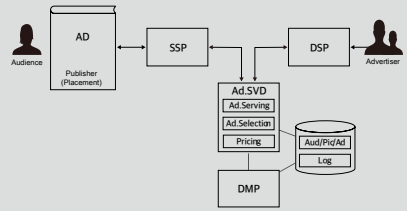
광고 시스템 개요

카카오의 광고 플랫폼에서 사용하는 기계학습 알고리듬을 이해하기 위해서 먼저 광고 시스템이 어떻게 구성되고 어떤 순서로 광고가 집행되는지를 간략히 알아보도록 하자.

[광고 집행 흐름]

1. 광고주는 DSP(demand side platform)를 통해 광고 소재 창작, 타게팅 조건, 입찰 방법 등을 등록한다.
2. 오디언스(audience)*3가 웹 페이지에 접속하면 광고 매체를 관리하는 SSP(supply side platform)를 통해서 광고를 요청한다.
3. 광고 요청을 받은 광고 서버는 지면 정보를 받아와서 카카오 내부에서 분석, 저장한 오디언스/지면/광고 정보를 기반으로 현 상황에 가장 적합한 광고를 선택해서 SSP에 전달한다. 추가로 프리퀀시 캐핑(frequency capping), 어뷰징 체크(fraud detection), 예산 관리 등을 통해서 광고 노출을 제어하고, 악속된 액션(노출, 클릭, 전환 등)에 맞게 과금한다.
4. 이상의 모든 과정은 DMP(data management platform)를 통해서 기록(log)으로 남긴다.

[그림 1] 광고 집행 플로우와 시스템 구성



광고 시스템은 일반 서비스에 활용되는 CMS(content management system)와 크게 다르지 않으나, 역할에 따라서 DSP, SSP, 애드서빙, DMP 등으로 구분한다.

[광고 시스템]

1. DSP(demand side platform)

DSP는 광고주와의 접점이다. 즉, 카카오가 관리하는 모든/특정 지면에 광고를 노출하려는 광고주가 접속해서 1) 광고의 목적을 설정하고 (캠페인 등록), 2) 메시지가 전달될 타겟 오디언스, 노출될 지면, 집행 기간, 예산 및 입찰 조건을 설정하고 (광고 그룹 등록), 3) 오디언스에게 전하고 싶은 메시지와 형식을 설정하고 (광고 소재 등록), 그리고 4) 최종적으로 집행된 광고의 효과를 확인하는 보고서 기능을 제공한다. 특히, 니즈가 명확한 검색광고가 아닌 일반 배너광고(display ad)에서 오디언스 타게팅 기능이 중요해지고 있다. 오디언스 타게팅은 오디언스의 기본 데모그래픽 (성별, 나이대, 거주지 등)*4, 카카오 내 활동을 분석해서 추정한 관심사 등으로 타겟 그룹을 분화한다. 현재 카카오는 마이다스(Midas)와 클릭스(Clix)라는 DSP를 제공하고 있으며, 현재 모멘트(Moment)라는 새로운 통합 DSP도 CBT(closed beta test) 중이다.

2. SSP(supply side platform)

DSP가 광고주와의 접점이라면, SSP는 카카오 광고를 노출하는 매체 또는 발행자(publisher)와의 접점이다. 카카오 광고는 카카오 내부 지면 (카카오톡, 카카오 스토리, 다음앱, 다음PC포털 등)과 외부 네트워크 지면 (카카오와 다음 서비스 이외의 외부 언론사 홈페이지, 독립 개발자의 앱 등)에 노출된다. 이런 모든 지면을 등록해서 광고를 제공할지 여부를 판단, 관리하는 것이 SSP의 역할이다. 그리고 최근에는 프로그래매틱 바잉(programmatic buying) 또는 실시간 비딩(real-time bidding, RTB) 등이 애드테크(AdTech)의 트렌드인데, 이런 AdX(AdExchange)의 광고 중계(mediation)도 SSP의 역할이라 할 수 있다. 이미 카카오는 크리테오, GDN(Google Display Network) 등의 외부 DSP가 제공하는 광고도 애드핏(AdFit, 카카오 SSP)를 통해서 내부 지면에 노출하고 있다.

3. AdServer
애드서버 또는 애드서빙(AdServnig) 시스템은 DSP와 SSP에 별도로 존재해서 광고 요청에 가장 적합한 광고를 선별하는 기능과 어떤 매체에 광고를 제공할지를 조정하는 기능을 한다. 편의상 하나의 가상 애드서버가 있는 것으로 도식화했다. 크게 1) 광고를 매체(또는 SSP)에 제공하는 애드서빙(AdServing), 2) 현재 맥락에서 접속한 오디언스 및 매체에 가장 적합한 광고를 선택하는 애드 셀렉션(Ad Selection), 그리고 3) 오디언스가 약속한 액션을 취했을 때 얼마의 금액을 광고주에게 요청할 것인가를 정하는 과금액 산정(pricing) 기능을 담당한다. 그외에 광고주의 일예산 관리 (예산 분배 및 광고 중지), 오디언스의 광고 피로도를 고려한 프리퀀시 캐핑, 광고 어뷰징 제어, 과금(billing), 그리고 실시간 광고중계 등의 기능도 담당한다. 이번 글은 애드 셀렉션(Ad Selection)을 중심으로 더 자세히 설명할 예정이다.

4. DMP(data management platform)
DMP는 광고 집행에 필요한 모든 데이터와 집행 중에 발생하는 모든 이벤트 로그를 저장/관리하는 시스템이다. DSP, SSP, 애드서버 등에 DMP 기능이 분산돼있으나 편의상 하나의 가상 시스템으로 그렸다. DSP를 통한 광고주 및 광고 데이터와 SSP를 통한 매체 데이터 뿐만 아니라, 카카오가 적법하게 수집한 데이터로 추청/분석한 오디언스 정보도 관리한다. 그리고 광고를 집행 하면서 발생하는 광고 요청, 노출, 클릭, 전환 및 빌링 로그 등을 저장한다. 특히 최근에는 광고에서 전환(conversion)이 중요해져서 CTS(conversion tracking system) 및 MAT(mobile app tracking) 등 외부 데이터를 관리하는 역할도 한다.

5. DR(data repository)
데이터 저장소는 DSP와 SSP에 등록된 모든 데이터와 DMP를 통해서 수집한 모든 정보와 로그를 저장해서 요청시에 빠르게 제공해주는 곳이다. 특히 애드 셀렉션에서 30~40ms 내에 광고를 선별해 제공하기 위해서 빠르게 데이터를 찾아서 안정적으로 전달해줘야 한다.

추천 시스템과 광고랭킹 시스템

카카오 AI 리포트 6월호에 카카오의 추천 엔진인 토로스(TOROS)가 소개되었다. 어떤 이는 광고랭킹도 광고 소재를 추천하는 것이니 토로스를 광고랭킹에 바로 적용하면 되는 것 아니냐?라고 반문할지 모른다. 광고랭킹이 일반 추천과 다른 점이 몇 가지 있다.

첫째, 추천 시스템은 보통 추천되는 아이템 또는 컨텐츠의 순서만 중요하고 소위 Top-N으로 명명되는 상위의 N 개의 아이템만 잘 추려내면 된다. 반면 광고랭킹은 추천 순서 뿐만 아니라, 정확한 추천 점수가 필요하다. 보통 pCTR(predicted click-through rate) 또는 이를 기반으로 한 eCPM(effective cost per mille)[5]을 정확히 계산해야 한다. eCPM에 따라서 광고의 노출 여부가 결정될 뿐만 아니라, 노출/클릭/전환에 따른 과금액이 산정된다. 과금액 산정이 부정확 하다면 카카오 입장에서는 제대로 된 매출을 기대하기 어렵고, 광고주 입장에서는 과대계상된 금액을 지불할 우려가 있다.

둘째, 추천 순서와 함께 과금액을 함께 결정해야 한다. 추천된 컨텐츠를 사용자가 조회했다고 해서 과금하지 않지만, 광고는 (CPC 상품인 경우) 노출된 광고를 오디언스가 클릭하면 할인된 CPC(cost-per-click)금액을 광고주에게 청구한다. 광고주가 제시한 입찰가(BA, bidding amount) 대로 과금하는 최고호가경매(first price auction)이면 큰 문제가 없지만, 보통 광고 시장에서는 더치경매(Dutch auction), 또는 Vickery-Clarke-Groves auction(VCG)으로 알려진 세컨프라이스경매(second price auction)에 기반해서 과금액을 산정한다. 이때 eCPM 또는 광고 시스템 별로 지정된 랭킹점수(engagement score)의 비율에 따른 할인율을 적용한다.

셋째, 높은 수준의 시스템 안정성이 요구된다. 재차 강조하지만 광고는 회사의 매출과 직결돼서 장애 허용성(fault tolerance)이 보장돼야 하고, 지연(latency)이 최대 100ms 이내의 빠른 응답으로 타임아웃(timeout)도 없어야 한다. 단순히 서버의 수를 늘려서 처리량(throughput)을 높이고 장애 백업을 지원한다는 의미만은 아니다. 소프트웨어 측면에서 알고리듬의 복잡도를 낮추고 문제가 발생했을 때 역추적도 가능해야 한다는 의미다. 딥러닝이 여러 분야에서 두각을 나타내고 있지만, 광고 시스템에서 여전히 보수적인 이유도 속도와 안정성, 그리고 사후 문제 재현 및 해결 가능성이라는 측면이 있다. 시시각각으로 변하는 사용자들의 반응을 실시간으로 반영하기 위해서 온라인으로 학습이 가능한 알고리듬이 필요하다.

마지막으로, 시스템 이슈는 아니지만, 광고랭킹은 사용자(오디언스)와 광고주 양쪽을 동시에 만족시켜야 한다. 극단적으로 얘기하면, 사용자 만족을 위해서 광고를 전혀 내보내지 않거나 광고주를 위해서 무조건 많은 광고를 내보내면 된다. 그렇게 되면 매출을 올리지 못하거나 광고 때문에 사용자들이 서비스를 떠나게 된다. 어떤 광고를 누구에게 언제 내보내야할 것인가를 미세하게 잘 조절할 수 있느냐가 광고 서비스의 승패를 좌우한다.

pCTR 예측

광고랭킹 알고리듬을 설명하기에 앞서 온라인 광고의 발전 과정을 간략히 살펴볼 필요가 있다. 인터넷 초기에는 배너 광고를 일정한 시간동안 노출시켜주는 CPT(cost per time) 상품이나 1,000회 노출당 과금하는 CPM(cost per mille) 상품이 중심이었다. 일정 시간동안 노출 회수를 보장하면 되기 때문에, 시간당 과금액 또는 노출당 과금액으로 광고를 정열한다. 이후 검색광고가 등장하면서 클릭당 과금(cost per click, CPC) 상품이 등장하고, 배너 광고도 CPC로 판매하는 성과형 광고가 등장했다. CPC 상품은 eCPM으로 환산해서 랭킹하기 때문에 정확히 pCTR을 예측하는 방법이 핵심이다. 최근에는 더욱 직접적인 전환을 기준으로 과금하는 CPA(cost per action) 상품이 등장하고, CPC 상품과 유사하게 예측전환율 (pCR)[6]로 eCPM을 계산한다. 현재 CPC 상품을 중심으로 광고 시장이 형성돼있으므로 pCTR 예측 방법을 중심으로 설명한다.

과거자료 CTR(historical CTR, hCTR)

pCTR을 예측하는 가장 쉬운 방법은 과거자료(historical data)를 집계하는 거다. 예를 들어, 모든 '(추정) 20대 남성'의 지난 일주일동안 광고별로 노출과 클릭 회수를 수집한다. 광고 A는 100,000회 노출에 1,500회 클릭이 발생했고, 광고 B는 70,000회 노출에 1,200회 클릭이 발생했다면 광고 A의 pCTR = 1,500 / 100, 000 = 1.5%이고 광고 B의 pCTR = 1,200 / 70,000 = 1.7%가 된다. 만약 광고 A와 B 모두 입찰가(BA)를 2,000원으로 설정했다면, 다음 번에 (추정) 20대 남성 U가 접속했을 때 eCPM(A) = 0.015 * 2,000 * 1,000 = 30,000원, eCPM(B) = 0.017 * 2,000 * 1,000 = 34,000원이 되어, eCPM이 높은 광고 B를 노출시키고 클릭이 발생하면 1,765 (= 2,000 * 30,000 / 34,000) 원을 과금한다. 과거자료만 누적, 집계하면 되는 가장 간단한 방법이다. 전형적인 MAB(multi-armed bandit) 또는 탐색-탐험(exploration-exploitation) 문제/접근법이다. 뒤쪽에서 언급할 톰슨 샘플링(Thompson sampling)과 결합해서 진화된 방식으로 해결할 수 있으나, 단순 과거자료 집계 방식은 한계가 명확하다.

첫째, 모든 오디언스 그룹, 광고 소재, 그리고 게재 지면 조합별로 CTR 데이터를 수집해야 한다. 단일 지면에서, 성별과 연령대로 구분하고, 10여 개의 광고 소재만 존재한다면 200개 미만의 조합으로 노출/클릭 데이터를 집계하면 되겠지만, 카카오의 경우 1,000개가 넘는 게재 지면을 가지고 있고 오디언스 피쳐(feature)는 최소 수십만 차원에 이르고, 광고 소재의 수도 10,000개가 훨씬 넘는다. 그리고 매일 새로운 광고 소재가 등록되고 시의성 없는 소재가 제외된다. 오디언스와 광고 소재만 고려하더라도 수천만 개의 세그먼트(segment) 별로 CTR 데이터를 집계해야 한다.

둘째, 강건한(robust) CTR 예측을 위해서 세그먼트 별로 일정 회수 이상의 노출이 보장돼야 한다. 예를 들어, 100회 노출에서 5회 클릭이 발생했을 때 CTR = 5%라고 추정한다면 신뢰할 수 없다. 세그먼트 별로 10,000회 정도 노출해서 517회 클릭이 발생한다면 그제서야 CTR = 5.17%라고 예측할 수 있다. 모든 세그먼트 별로 신뢰할 수 있는 수준의 CTR을 얻기 위해서 최소 회수의 노출을 보장하는 것은 트래픽이 많은 서비스에서도 부담이 된다.

셋째, hCTR은 요일 및 시간에 따른 변동성(fluctuation)을 뭉개버린다. 지난 일주일 데이터로 CTR = 5%를 예측하더라도 현재/미래의 트래픽에서 4%가 될지 6%가 될지 장담할 수 없다. 카카오도 초기에는 최소의 세그먼트로 hCTR을 적용했지만 예측력이 많이 낮아서 결국 폐기하고, CTR 예측모델을 만들어서 pCTR을 계산하는 걸로 선회했다. 반직관적이지만 실측치가 예측치보다 못하다.

로지스틱 회귀모델(logistic regression)

CTR의 정의는 노출 횟수(impressions) 대비 클릭 횟수(clicks)의 비율이다. 즉, CTR = clicks / impressions로 계산한다. 전체 노출 횟수에 대한 전체 클릭 횟수의 비율을 달리 표현하면, 1회 노출당 클릭될 확률을 구하는 것과 같다. 오디언스 U에게 광고 A를 노출했을때 오디언스가 그 광고를 클릭할 조건부 확률을 구하는 문제다(pCTR = Pr(click = 1 | impression)). 이런 문제에 적합한 해결책은 최대 엔트로피(maximum entropy)로 알려진 로지스틱 회귀모델(logistic regression)이다. 각 회사의 광고 플랫폼마다 사용하는 피쳐의 종류와 엔지니어링 방법만 조금씩 다를 뿐, 현존하는 대부분의 광고랭킹은 로지스틱 모델에 기반하다.

로지스틱 회귀모델을 간략히 설명하면 다음과 같다. 학습 데이터 세트 (xi, yi)가 주어졌을 때, xi는 0 또는 1을 갖는 d 차원의 성김 피처 벡터(sparse feature vector)이고, yi는 -1 또는 1을 갖는 목표값이다. 회귀식을 구할 때 실수값 x로 계산할 수 있지만, 광고랭킹에서는 보통 이산화(discretization)해서 바이너리 값을 갖게 한다. 카테고리 데이터도 모두 임의(dummy)변수로 만든다. 이때 d 차원의 가중치 벡터 w를 찾는 것이 목적이다. 샘플 x가 y = 1로 예측할 확률은 다음과 같다.

$$Pr(y = 1 \mid x, w) = \frac{1}{1 + exp(-w^T x)}$$

이때 가중치 벡터 w를 구하기 위해서 아래의 부정 로그-확률 손실(negative log-likelihood loss)함수를 계산한다.(L_2 regularization을 사용함)

$$\min_w \frac{\lambda}{2} \| w \|^2 + \sum_{i=1}^{n} log(1+exp(-y_i w^T x_i))$$

위의 손실 함수는 볼록성(convex) 최적화 문제로 보통 기울기(gradient) 기반의 최적화 방식으로 쉽게 해결된다. 충분히 많은 학습 데이터 세트가 있다면 모델 가중치 w는 쉽게 구할 수 있다. 그런데 배치(batch) 최적화로 가중치를 계산하는 것은 시간이 오래 걸려서 최신의 피드백 (클릭/전환 데이터)을 바로 반영하지 못한다. 그래서 확률적 확률적경사하강(stochastic descent gradient, SGD) 방식의 온라인 최적화 방식을 주로 이용한다[7].

피쳐 데이터와 차원 축소(feature engineering)

온라인 최적화 방식(FTRL-Proximal)을 설명하기 전에 카카오에서 사용하는 각종 피쳐 데이터(feature data)와 전처리 과정부터 알아보자. 정확한 pCTR을 예측하기 위해서 다양한 데이터를 활용하고 있다. 구체적으로 어떤 데이터를 어떻게 수집하고 활용하느냐는 회사마다 갖는 노하우(know-how)이므로 자세히 다루지 못함을 양해바란다.

첫째, 오디언스의 데모 정보를 추정해 사용한다. 카카오에서의 다양한 활동을 분석해서 베이지 추론(bayesian inference)으로 성별과 연령대를 추정한다. 주 활동지 등의 지역은 IP-지역 매핑 데이터(IP mapping data) 등을 활용해서 추정하고 있다.

둘째, 다음검색을 통해 유입된 검색어 이력이나 소비한 카페나 스토리채널의 메타데이터를 분석해서 오디언스의 관심사로 매핑하고 있다.

광고 소재의 경우 광고주ID, 개별 소재ID, 소재의 크기와 타입, 소재 등록/검수 시에 지정한 카테고리 등을 이용하고 있다. 소재의 URL, 텍스트 또는 이미지 등에서 객체를 추출해서 사용하는 것도 테스트 중이나 앞서 해결할 문제들이 있어서 아직 상품에 적용하지는 않았다. 그리고 게재지면(매체)의 지면ID와 publisherID, 카테고리, 그리고 지면별 반응률 등을 활용하고 있다. 마지막으로 요일과 시간 등의 맥락 정보도 활용한다.

개별 오디언스 피쳐 벡터(수십만 차원), 광고 피쳐 벡터(수천 차원), 지면 피쳐 벡터(수천 차원)만으로도 데이터 차원이 매우 커서 연산이 거의 불가능하다. 그런데 이런 피쳐들의 상호작용(interaction) 효과를 반영하기 위해서 이종 피쳐 간의 데카르트 곱(cartesian product)을 취한다. 그렇게 되면 적게는 수백만 차원, 많게는 수억 차원의 피쳐 벡터가 만들어진다. 로지스틱 회귀모델에 L_1 정규화(regularization)를 적용해서 피쳐 선택(feature selection) 기능을 넣더라도 연산이 불가능하다. 그래서 최소한의 정보 손실을 감내하더라도 연산이 가능하게 데이터의 차원을 축소하는(data reduction) 기술이 필수다. 저차원의 데이터라면 특이값 분해(singular value decomposition, SVD)나 주성분분석(principal component analysis, PCA)으로 차원 축소가 가능하지만, 대용량 데이터는 전체를 메모리에 올려서 계산해야 하기 때문에 현실성이 적다. 그래서 야후!(Yahoo!)와 크리테오(Criteo)에서는 해싱 트릭(hashing trick)을 이용하고, 페이스북은 그레디언트 부스팅 머신(gradient boosting machine, GBM)을 이용한 부스티드 결정 트리(boosted decision tree)를 사용한다. 구글은 블룸 필터(Bloom filter)를 이용해서 새로운 피쳐가 기존 모델에 추가되는 것을 최소화하고 있다.

카카오에서는 현재 클러스터링 기법을 활용 중이다. Cartesian product된 모든 피쳐로 클러스터링하는 것이 현실적으로 불가능해서, 유사한 오디언스나 지면을 묶는 식으로 추상화한다. 축소된 오디언스, 지면, 그리고 광고 피쳐를 cartesian product해서 사용한다. 수십만 차원의 오디언스 데이터로 클러스터링하는 것도 부담되기 때문에, 우선 랜덤프로젝션이나 해싱트릭으로 차원을 축소한 후에 클러스터링을 하면 어느 정도의 정보 손실은 있지만 큰 부담없이 연산할 수 있다. 차원 축소에서 클러스터링이 좋은 이유는 초기 구축 비용이 많이 들어가지만, k개의 중심값(centroids)만 계속 업데이트하면 되기 때문에 데이터 연속성을 보장하면서 연산량은 별로 크지 않다는 점이다.

위에서 피쳐의 차원을 축소하는 방법을 기술했는데, 샘플의 개수를 줄이는 방법도 고려할 수 있다. 실제 CTR을 확인하면 1% ~ 10%로 매우 낮다. 즉, 클릭으로 연결된 노출의 비율이 매우 낮다는 의미다. 그래서(x, y) 쌍에서 90% 이상이 y=-1인 샘플이다. 그래서 일정 비율의 y=-1 샘플만으로 로지스틱 회귀모델을 구축한 후에 보정하는 방법을 사용한다. y=1인 샘플은 상대적으로 매우 적기 때문에 모든 샘플을 학습 데이터로 사용한다. 그외에도 랜덤 샘플링(random sampling)을 포함해서 코어셋(coresets)이나 스케치(sketch) 등의 기법으로 핵심 샘플만 남겨서 클러스터링(clustering)의 초기 중심점을 구하는 방식도 고려할 수 있다.

광고 플랫폼을 제공하는 회사들은 많이 있다. 대부분의 회사들은 다음에 소개할 FTRL(follow the regularized leader) 기법으로 로지스틱 회귀모델을 만들어서 pCTR 예측모델로 사용하고 있다. 예측모델과 방법론은 대동소이하다는 거다. 그럼에도 회사마다 예측 품질에 차이가 나는 것은 각 플랫폼마다 어떤 피쳐를 사용하고 있으며 (또는 얼마나 다양하고 많은 오디언스 정보를 수집하고 있으며) 그런 피쳐들을 어떻게 전처리하느냐에 달려있다. 페이스북은 방대한 소셜 그래프(social demography)를 활용하고, 구글은 사용자의 니즈가 명확한 검색어를, 그리고 크리테오는 광고주의 랜딩URL에 심어둔 리타게팅 스크립트(re-targeting script)를 통해 오디언스 정보를 수집, 정교화한다. 전처리에서는 페이스북은 결정 트리(decision tree)로 전처리를 해서 3% 이상의 개선 효과를 얻었지만, 다른 여러 휴리스틱(heuristic)을 적용했을 때는 추가 개선을 별로 얻지 못했다고 한다[8]. 또 크리테오는 해싱 트릭을 사용하지만, 구글의 실험 결과에서 해싱 트릭은 별로 효과가 없었다고 한다[9][10].

온라인 모델 업데이트, FTRL-Proximal

여기까진 FTRL을 설명하기 위한 배경지식이다. 가장 최근에 나온 버전이 FTRL-Proximal인데, 이름에 사용된 용어를 알면 그 의미를 쉽게 알 수 있다. FTRL은 follow the regularized leader의 약자다. 최초에는 FTL이 소개됐고, 후에 정규화(regularization)가 포함된 FTRL이 됐고, 여기에 근접치(proximal)가 추가된 형태로 진화했다.

왜곡이나 편향이 없다면 학습 데이터의 양이 많을수록 더 정확하고 강건한 예측모델이 만들어지는 것은 기계학습의 기본이다. 하지만 학습 단계에서 전체 모집단(population)을 잘 설명하는 모든 데이터를 얻는 것은 현실적으로 불가능하다. 광고에는 새로운 유형의 오디언스가 계속 등장하고, 새로운 광고주와 광고 소재가 등록 (또는 폐기)되고, 새로운 성향의 매체 지면이 추가되는 것이 빈번하다. 초기 모델을 학습하면서 그런 변동성을 모두 파악/반영하는 것은 불가능하다. 그리고 정상 학습 데이터도 양이 많아지면 학습하는데 많은 메모리와 시간이 필요하다. 참고로 카카오는 매일 수억 건의 광고 요청을 받아서 광고 소재를 노출시키고 수천만 건의 클릭 등의 포스트액션 데이터를 수집한다. 학습을 위해서 일주일, 한달 치의 데이터를 모두 사용한다면 많은 서버를 투입해도 수시간이 필요하다. 데이터의 변동성과 학습 효율을 고려해서 분할 정복(divide & conquer) 방식을 이용해 시간 단위로 분할 학습과 통합 모델 구축 전략을 사용한다. 이 전략의 가장 최신 버전이 FTRL-Proximal이라고 보면 된다.

일반적인 최적화 방식인 온라인경사하강(online gradient descent, OGD)은 현재 타임프레임 t에서의 가중치 wt와 기울기 gt를 이용해서 t + 1에서의 가중치를 구하면 다음과 같다.

$$w_{t+1} = w_t - \eta_t g_t$$

여기서 g_t 앞에 있는 값은 아래와 같은 단조감소(non-increasing)하는 학습 파라미터(parameter)다.

$$\eta_t = 1/\sqrt{t}$$

FTRL-Proximal은 대신 아래의 식으로 가중치를 업데이트한다.

$$w_{t+1} = argmin_w(g_{1:t} \cdot w + \frac{1}{2}\sum_{s=1}^{t}\sigma_s \| w - w_s \|_2^2 + \lambda \| w \|_1;$$

$g_{1:t} \cdot w$ $\quad \| w - w_s \|_2^2$ $\quad \| w \|_1$

위의 식은 크게 세 부분으로 나뉘어있다.

첫째, FTL(follow the leader)를 표현한 것으로, t 시간까지 가장 작은 손실을 갖도록 한 leader(gradient)들을 취해서 손실의 근사치를 구한다.

둘째, 새로운 가중치가 이전에 사용했던 가중치들에서 큰 변동이 없도록 제약하는 근접치(proximal) 파트다. 이런 강한 볼록성(convexity)을 추가함으로써 알고리듬의 안정성을 높인다.

마지막 L_1정규화(regularization) 파트이다. 이들을 하나로 합친 것이 FTRL-Proximal인데, 현재까지의 최고 서브그레디언트(subgradient)의 궤적을 쫓으며 근사한 손실을 최소화하고 규칙화와 근접치를 통해서 모델의 급격한 변동성을 막는 안정된 모델을 구축한다. 온라인 최적화 방식이 모든 학습 데이터로 한꺼번에 계산하는 배치(batch) 방식보다 정확도는 다소 낮을 수 밖에 없다. 그럼에도 빠른 학습/업데이트 시간과 최소 리그렛 바운드(regret bound)를 보장하는 최적화 방법 중에 FTRL-Proximal이 현재의 최신기술(state-of-the-art, SOTA)이다. FTRL보다 더 나은 예측 정확도를 보여주는 알고리듬들이 계속 소개되고 있지만 제품 단계에서 적용하는 데는 아직 한계가 있다. FTRL-Proximal에서 규칙화 와 근접치 부분을 조금 변경한 FOBOS, AOGD, RDA 등의 방식도 함께 연구됐지만, 현재까지는 FTRL-Proximal이 가장 안정된 성능을 보여준다[11].

피크 타임에는 초당 수십만 건의 광고 요청이 들어오고, 모든 요청은 30~40ms (길어도 100ms) 내에 가장 적합한 광고를 선택해서 내보내야 한다. 그래서 광고 랭킹에 사용하는 알고리듬은 빠른 연산과 안정된 서빙이 가능해야 한다. 그래서 빠른 알고리듬에 더해서 사용하는 피쳐의 수도 최소화하려는 거다. 앞서 설명한 차원 축소가 정확한 예측 모델 구축 뿐만 아니라, 안정된 서빙(serving)을 위해서도 중요하다. 보통 예측 시간보다 모델 구축 시간이 오래 걸린다. 만약 딥러닝을 이용한 방식이 더 정확한 예측력을 제공한다면 오프라인에서 딥러닝 모델을 구축하고 온라인에서 딥러닝 기반으로 예측만 하면 될 것 아니냐?라고 반문할 수 있다. 합리적 추론이지만, 변동성이 심한 인터넷 트래픽 환경에서는 모델의 새로움(freshness)을 유지하는 것이 중요하다. 즉, 하루 이틀동안 학습한 모델을 일주일, 한달동안 계속 사용할 수가 없고, 수분에서 1시간 주기로 새로운 데이터로 모델을 계속 업데이트해줘야 한다. 딥러닝 모델이 예측력이 더 낫다는 것이 증명되고 효과적인 업데이트가 가능해지면 광고 랭킹시스템도 진일보할 것이라고 본다.

기타 이슈

이상으로 광고랭킹에 사용하는 피쳐 작업과 FTRL 알고리듬 기반의 pCTR 예측모델 학습을 알아봤다. 광고랭킹에서 pCTR을 예측하는 것이 가장 중요한 작업이지만, 이외에도 해결해야할 데이터 및 기계학습 문제들이 여럿 있다. 이번 글에서는 모든 기술적 디테일과 해결책을 설명하기 보다는 개념적으로 문제와 해결 방식만을 간략히 소개하려고 한다. 자세한 것은 참고 논문을 읽기 바란다.

톰슨 샘플링(Thompson sampling)

앞서 hCTR을 소개하는 단락에서 톰슨 샘플링을 언급했다. 안정적인 hCTR을 얻기 위해서 충분한 노출 기회가 필요한데, 현실적으로 모든 세그먼트에게 많은 노출 기회를 균등하게 제공할 수 없다. 그래서 사후확률(posterior)을 고려해 더 잠재력 높은 세그먼트에 더 많은 트래픽을 몰아줘서 빠르게 수렴하게 한다. 예측 모델을 이용한 pCTR 계산에서도 톰슨 샘플링을 사용한다. 신규 등록된 광고 소재의 정확한 pCTR을 예측하기 위해서 다양한 트래픽에서 충분한 노출과 클릭 데이터를 수집해서 학습한다. 초기 빈약한 데이터에서도 pCTR이 높게 나온다면 자연스럽게 eCPM 입찰 경쟁에서 이겨서 노출 기회를 얻고, 실제 많은 노출에서 실측한 CTR이 pCTR보다 낮게 나오면 자연스레 온라인 업데이트를 통해서 pCTR이 낮게 조정된다. 반대 상황에서 낮게 예측된 부정확한 pCTR은 해당 광고 소재의 노출 기회를 박탈해서 정확한 pCTR 모델의 구축이 불가능하다. 예를 들어, 과거의 다른 광고들을 집행하며 얻은 통찰에 따라서 예측된 CTR의 잠재력이 15%인데, 초기 1,000회 노출에서 120회의 클릭을 받아서 pCTR이 12%로 예측하도록 초기 모델이 만들어졌다면, 잠재력이 더 크기 때문에 인위적으로 더 많은 노출 기회를 제공해서 실제 CTR이 잠재 CTR인 15%가 맞는지 아니면 초기 실측치인 12%가 맞는지를 확인하면서 예측모델을 조정하게 된다[12].

광고 품질 지수

많은 사람들이 광고를 싫어한다. 전세계적으로 PC에서 AdBlock을 설치한 브라우저 수가 2억개를 넘었고, 모바일에서는 약 4억개의 AdBlock이 설치됐다는 조사 결과가 이를 입증한다[13]. 그렇기 때문에 광고 플랫폼 사업자는 오디언스에게 더 적합한 광고를 제공해 서비스 경험을 해치지 않게 하려 노력한다. 그럼에도 부적절한 소재와 형식의 광고가 노출되는 것을 완전히 사전에 차단할 수는 없다. 때로는 모든 광고를 싫어하는 오디언스도 존재한다. 그래서 다양한 형태의 오디언스 피드백을 모아서 광고 기피 사용자에게 광고를 덜 보여주기도 하고, 오디언스별 선호도에 따라서 연관성이 낮은 광고 소재를 사전에 필터링하는 알고리듬도 적용한다. 명시적으로 '이 광고 보기 싫어요'와 같은 피드백도 수집하지만, 우발적으로 클릭된 데이터를 향후 모델 학습에서 배제하는 등의 네가티브 피드백(negative feedback)을 광고 품질 지수에 포함시켜서 활용한다.

CVR(conversion rate) 예측

과거의 온라인 광고는 단순히 브랜드 홍보를 위해서 광고 노출 기회를 얻거나 광고를 통해서 광고주의 홈페이지로 사용자를 유입시키는 것이 목적이었다. 하지만 최근에서 광고 노출 단계에서부터 광고의 목적, 즉 전환을 고려해서 광고를 집행하고 있다. 광고의 목적은 모바일 앱을 설치하고 실행한다거나 쇼핑몰에 가입해서 물건을 구매한다거나 자동차 시승 이벤트에 응모를 한다는 등으로 다양하다. 이런 다양한 목적 별로 적합한 오디언스가 다르기 때문에, 오디언스 별로 광고의 전환율을 예측하는 것이 중요해졌다. CVR (클릭 후 전환율 = Conversions / Clicks)의 예측도 앞서 설명한 로지스틱 회귀모델과 FTRL 알고리듬으로 가능하다. 하지만, 노출과 클릭 데이터에 비해서 전환 데이터는 수집하기 어렵고, 외부 CTS/MAT(conversion tracking system / mobile app tracker)를 통해 수집된 데이터도 유효성 검증이 필요하다. 유효한 전환 데이터를 제대로 수집하더라도 노출, 클릭에 비해서 전환수가 극히 적기 때문에 신뢰도 높은 전환 예측 모델을 만들기도 어렵고, 광고 클릭 후 전환 발생까지의 시간 차가 커서 앞서 설명한 방식을 그대로 적용하는데 어려움이 있다. 지연된 피드백 모델(delayed feedback model)과 같은 알고리듬도 제시됐지만 제품화를 위해서 넘어야할 장애물이 많다[14].

어뷰징 처리(fraud detection)

광고 서비스는 현금이 거래되기 때문에 다양한 형태의 어뷰징(abusing)이 이뤄진다. 경쟁사의 광고비를 빨리 소진시키기 위해서 경쟁사의 광고를 거짓으로 클릭하는 경우 (CPC 상품인 경우)도 존재하고, 매체를 제공하는 발행자(publisher)가 수익을 더 얻기 위해서 자신의 사이트에 노출된 광고를 과하게 클릭하는 경우도 있다. 이런 류의 단순 어뷰징은 간단한 로직으로 해결할 수 있지만, 계속 지능화되기 때문에 100% 잡아내기가 어렵다. 어뷰징은 아니지만 사용자의 실수로 여러 번 클릭해서 이중으로 과금하는 경우도 있다. 이런 어뷰징이나 우발적 실수에 의한 과금은 분석을 통해서 비과금 처리한다. 그리고 왜곡 데이터가 모델 학습에 사용되지 않도록 차단하는 것도 중요하다.

인수분해(factorization)

최근 FTRL보다 더 나은 성능을 보이는 알고리듬이 소개됐다. 최초의 아이디어는 행렬 인수분해(matrix factorization)과 SVM(support vector machine)의 개념을 결합한 인수분해기계(factorization machines)였는데, 향후 개별 피처(feature)들의 메타정보(field)를 모델에 반영해서 FFM(Field-aware factorization machine)으로 발전했다. FTRL을 SOTA라고 설명했지만, pCTR 예측에서 FFM이 로직스틱 모델보다 3~4% 이상 정확도가 높다고 보고됐다. 하지만 로지스틱 모델과 달리 FFM은 가중치를 잠재 벡터(latent vector)화했기 때문에 연산량과 메모리 사용량이 더 높은 단점이 있다. 하지만, 최근 논문에서 다양한 시스템 튜닝(system tuning)을 통해 실제 광고 서빙에 사용하는데 큰 지장이 없다는 것을 보였기 때문에 카카오에서도 어떻게 제품에 적용할지 고민 중에 있다[15,16].

테스트와 제품

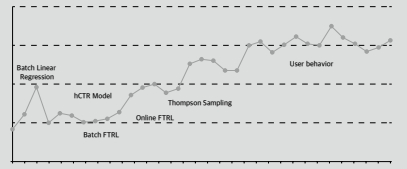
많은 애드테크(AdTech) 연구자들이 계속 새로운 문제와 해결책을 소개하고 있고, 카카오 내에서도 새로운 개선책을 계속 발굴하고 있다. 가장 좋은 것만 취사선택해서 시스템에 바로 적용하면 매출이 극적으로 오를 것 같지만 현실은 그렇지 않다. 우선 개별 아이디어가 실제 카카오 광고에서 유효한지를 판별해야 하고, 앞서 강조 했듯이 제품 단계로 확장(scale up)하는 과정이 필요하다. 그래서 새로운 아이디어의 구현과 테스트를 상시 진행한다. 구현 단계에서의 유닛 테스트를 제외하고, 테스트는 크게 3단계로 이뤄진다. 고정된 테스트 데이터를 이용해서 이전의 알고리듬들과의 성능 차이를 1차로 확인한다. 개선 효과가 있다고 판단되면 최근 일주일에서 한달동안 누적된 실제 집행 로그를 가져와서 현재 적용된 알고리듬과 새 알고리듬을 비교하는 오프라인 테스트를 거친다. 오프라인 테스트를 통과했다면 실제 트래픽에서 5~10%정도를 별도의 꾸러미(bucket)로 설정해 온라인 버킷 테스트 (또는 A/B 테스트)를 거친다. 온라인 테스트에서 유효성이 검증됐다면 버킷의 양을 차츰 늘리거나 전체 트래픽에 적용한다. 알고리듬 개선으로 매출 증대하는 것이 목적이지만, 기존 환경에 익숙한 사용자들이 느낄 충격을 최소화하기 위해서 다양한 테스트 과정과 점진적 트래픽 확대라는 전략을 취한다.

길게 설명하지는 않겠지만 이외에도 다양한 문제들이 있다. 개별 광고 소재의 과금액을 결정하고, 특히 CPM이나 CPC 등의 다른 가격 정책의 상품들이 혼재된 경우 이들 사이의 가격을 조정하는 프라이싱 문제가 있다. 개인별 네가티브 피드백을 분석하듯이 개인별 광고 피로도를 측정해서 같은 소재의 광고를 몇 회까지 보여줄 것인가를 정하는 프리퀀시 캐핑 문제도 있다. 광고주가 설정한 타겟에 따라서 특정 오디언스에게만 광고를 보여주거나 배제하는 것과 유사한 고객을 찾아서 광고를 부스팅하는 등의 개인화 문제도 알고리듬으로 해결해야 한다. 실측 CTR과 예측 CTR 간의 간극을 조정하는 캘리브레이션(calibration)도 공통 문제다. 최근에는 이른바 oCPM (optimized CPM)이라 불리는 상품이 등장해서 광고주가 입찰금액을 정하는 것이 아니라 알고리듬이 광고 목적과 연계해서 오디언스별로 다른 입찰금액을 제시하는 자동입찰 문제도 있다. 이상은 모두 DSP 관점에서의 문제였는데 SSP 측면에서의 최적화 문제도 함께 고민하고 있다. 이외에도 데이터 기반으로, 그리고 알고리듬으로 해결해야할 수많은 문제들이 있다.

카카오 광고의 성과와 향후 과제

이번 글에서 그동안의 노력과 성과를 모두 나열할 수는 없지만, 아래의 그래프를 보면 어느 정도 가늠할 수 있다. 2014년 7월에 카카오 스토리에 처음 배너 광고가 노출된 후로 약 3년 간의 eCPM의 월별 추이와 앞서 설명한 여러 알고리듬을 적용한 시기를 대략적으로 표시한 그래프다. 스토리 서비스의 개편이나 사용자 트래픽의 증감과 같은 다양한 외부 요인이 있었지만, 초기에 비해서 eCPM이 약 3배 증가했음을 확인할 수 있다. 어떤 알고리듬/개선책은 오프라인 테스트와 온라인 테스트에서 효과가 입증돼서 모든 트래픽에 적용했지만, 장기간의 추적을 통해서 결과적으로 효과가 없음이 나중에 밝혀져 시스템에서 빠진 경우도 있다. 대표적으로 1년 정도 유지하던 hCTR을 2016년도에 제외했다. 강력한 하나의 슈퍼 알고리듬이 온라인 광고의 승패를 결정하는 것이 아니라, 여러 적정 기술과 알고리듬들이 결합해서 광고 생태계를 이룬다.

[그림2] 피쳐 및 알고리듬 변경에 따른 eCPM의 월별 추이



많은 노력으로 광고랭킹 알고리듬을 개선하고 있지만 여전히 해야할 일이 많다. 가장 핫한 이슈인 '딥러닝을 광고 시스템에 어떻게 적용할 수 있을까?'도 해결해야 한다. 우선은 딥 오토인코더 (deep auto encoder) 같은 기술을 이용해서 피쳐 벡터의 차원을 축소하는 문제부터 적용해볼 수 있다. 또는 RNN/CNN 등의 기술을 활용해서 광고 소재 및 설명에서 유효한 객체를 찾아내고, 또 오디언스들이 소비한 컨텐츠에서도 유효 객체를 찾아서 매핑하는

것도 진행해야 한다. 로지스틱과 FTRL을 대체할 딥 아키텍처(deep architecture)도 가능하다고 본다. 딥러닝이 장기적인 과제라면, 앞서 설명한 FFM을 우선 프로덕션에 적용하는 것은 단기 과제다. 광고에서 1~2%의 개선도 전체 매출에서 큰 부분을 차지한다. 많은 광고의 CTR, CVR은 1% 미만이다. 어떤 광고는 이보다 더 낮은 값을 갖기도 한다. 하지만 만약 매우 낮은 CTR의 광고더라도 적절한 순간에 적절한 오디언스에게 전달된다면 그 오디언스에게는 가치있는 (CVR이 높은) 정보를 제공하는 셈이다. 9,999명에게는 불필요한 정보지만, 1명에게는 꼭 필요할 수도 있다. 그렇듯이 아주 희귀한 사건(rare event)에 부합하는 피쳐를 발굴해서 정확히 예측하는 모델도 개발해야 한다. SSP 관점에서의 최적화 문제, 정확한 CVR 예측을 위한 피쳐 발굴과 모델 학습, oCPM을 비롯한 자동 입찰 로직 개발, 트래픽 예측과 예산 분배, 광고의 효과를 정확히 평가하는 복수터치기여(multi-touch attribution) 문제 등 해결해야할 문제가 여전히 많다.

결언

지면 관계상 많은 디테일을 생략해 다소 어렵거나 재미없을 수도 있지만 광고 랭킹과 관련된 많은 문제들이 존재하고, 데이터 관점에서, 그리고 알고리즘으로 하나씩 그런 문제들을 해결해가는 것은 매우 흥미로운 과정이다. 이 분야에는 여전히 도전을 기다리는 수많은, 재미난 문제들이 존재한다. 앞서 보여준 성과보다 더 아름다운 우상향 그래프를 그려갈 수 있기를 기대한다.

***1** 참고 : 알파벳 회사 설명. https://abc.xyz/ ***2** 참고 : Netflix recommendations: Beyond the 5 stars http://kko.to/IUNr9VxEA ***3** 참고 : 광고에서 '사용자'를 audience라 부름. ***4** 참고 : 카카오는 개인정보 보호를 위해서 사용자의 실제 성별이나 연령을 사용하지 않고, 카카오 내에서의 다양한 활동을 추적해서 베이지 추론 등의 기법으로 데모 정보를 추정해서 사용한다. 그리고 k익명성을 고려해서 오디언스의 정보를 추상화 및 군집화해서 관리한다. 광고 효율은 다소 낮아질 수 있으나, 인터넷에서의 개인정보 보호라는 더 큰 원칙을 준수하려 노력한다. ***5** 참고 : eCPM = BA(billing amount) x pCTR(page click-through rate) ***6** 참고 : 전환율은 보통 CR(conversion rate) 또는 AR(action rate)라고 부르는데, 이는 노출당 전환 비율 (= #conversion / # impressions) 을 뜻한다. 논문 등에서 CVR이라는 용어도 사용하는데, 이는 포스트 클릭 전환, 즉 클릭된 회수 대비 전환 비율 (= #conversions / # clicks)을 계산한 것이다. 전환은 매우 희귀한 이벤트라서 보통 CR은 CTR와 CVR로 나눠서 구한다(CR = CTR * CVR). ***7** 참고 : How to implement logistic regression with stochastic gradient descent from scratch with Python http://kko.to/IUOe6qDp6 ***8** 논문 : He, Xinran, et al. "Practical lessons from predicting clicks on ads at facebook." Proceedings of the Eighth International Workshop on Data Mining for Online Advertising. ACM, 2014. ***9** 논문 : McMahan, H. Brendan, et al. "Ad click prediction: a view from the trenches." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013. ***10** 논문 : Chapelle, Olivier, Eren Manavoglu, and Romer Rosales. "Simple and scalable response prediction for display advertising." ACM Transactions on Intelligent Systems and Technology (TIST) 5.4 (2015): 61. ***11** 논문 : McMahan, Brendan. "Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization." Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. 2011. ***12** 논문 : Chapelle, Olivier, and Lihong Li. "An empirical evaluation of thompson sampling." Advances in neural information processing systems. 2011. ***13** 참고 : 2017 AdBlock report http://kko.to/IUOSMO3UQ ***14** 논문 : Chapelle, Olivier. "Modeling delayed feedback in display advertising." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014. ***15** 논문 : Juan, Yuchin, et al. "Field-aware factorization machines for CTR prediction." Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016. ***16** 논문 : Juan, Yuchin, Damien Lefortier, and Olivier Chapelle. "Field-aware factorization machines in a real-world online advertising system." Proceedings of the 26th International Conference on World Wide Web Companion. International World Wide Web Conferences Steering Committee, 2017.

"강력한 하나의 슈퍼 알고리듬이
온라인 광고의 승패를 결정하는 것이 아니라,
여러 적정 기술과 알고리듬들이 결합해서 광고 생태계를 이룬다"