[추천시스템] Practical Lessons from Predicting Clicks

Info.

- 본 문서는 Practical Lessons from Predicting Clicks on Ads at Facebook (Xinran He et al.) 논문을 요약 정리한 문서임.
- 목적1) 클릭 예측 모델링 구축의 실제 사례를 통해 변수 선정 및 하이브리드 모델 방식의 유용성을 이해한다.
- 목적2) 배민에 예측 모델 구축시, 응용 가능한 접근방법 및 아이디어를 도출하고 토의한다.
    - Introduction
    - Experimental Setup
    - Prediction Model Structure
    - Online Data Joiner
    - Containing Memory and Latency
    - Coping with Massive Training Data
    - Discussion
    - 생각해볼 문제들 in action
    - Next Step

작성완료

1. Introduction
    a. 배경
        i. 디지털 광고 시장의 성장에 따라 머신러닝 기법을 이용해 광고를 선별해서 노출하는 방식의 관심이 상승함 → 효율성 제고
        ii. 구글, 야후 및 Microsoft사의 선두 주자이며 Robustness 및 Adaptive 방식 그리고 대량의 데이터를 다룰 수 있는 시스템이 요구됨 → 효율성 확보
    b. 결론 요약
        i. Facebook의 경우 질의어와 광고가 연결되어 있지 않음. 따라서 Demo 및 관심사(interest) 정보를 이용해 유저 판별
        ii. 결과적으로 Decision Tree 및 Logistic Regression의 하이브리드 모델이 높은 성능을 보임
        iii. 많은 요인이 모델 성능에 영향을 미치지만 가장 큰 영향력을 가지는 것은 올바른 변수와 모델 유무(decision tree & logistic regression)임.
        iv. 최적화된 Data freshness, Learning rate schema, Sampling, Transform 시도는 올바른 변수와 모델에 비해 상대적으로 영향력이 낮음
2. Experimental Setup
    a. Normalized Cross-Entropy(NE)
        i. "The average log loss divided by what the average log loss per impression would be if a model predicted the background click through rate(CTR)"
        ii. yi∈{1,+1}
        iii. pi: 예측 클릭 확률
        iv. N: 샘플수
        v. p: 관측된 평균 CTR

$$NE = \frac{-\frac{1}{N}\sum_{i=1}^{n}\left(\frac{1+y_i}{2}log(p_i) + \frac{1-y_i}{2}log(1-p_i)\right)}{-(p*log(p)+(1-p)*log(1-p))}$$

            1.
        vi. 분자 → average log loss per impression (cross-entropy)
        vii. 분모 → normalized term, where p = 평균 추정치
        viii. NE의 값이 낮을수록 예측력이 좋음
    b. 정보이론 기초
        i. 출처 및 참고자료
            1. https://www.nist.gov/sites/default/files/documents/2017/11/30/nce.pdf
            2. http://blog.naver.com/PostView.nhn?blogId=gyrbsdl18&logNo=221013188633
            3. http://norman3.github.io/prml/docs/chapter01/6.html
            4. https://ratsgo.github.io/statistics/2017/09/22/information/

5. https://icim.nims.re.kr/post/easyMath/550
6. http://jaejunyoo.blogspot.com/2018/02/minimizing-negative-log-likelihood-in-kor-3.html

ii. Information → degree of surprise
1. 정보량(information)?

$$h(x) = -\log_2 p(x)$$

a.
2. 브라질 vs 중국의 축구 경기
   a. 브라질이 이기는 경우(99% 사전확률): -log P(x) = -log(0.99) = 0.01
   b. 중국이 이기는 경우(1% 사전확률): -log P(x) = -log(0.01) = 4.6

iii. Entropy → 불확실한 정도
1. 정보량의 기대값

$$H[x] = -\sum_x p(x) \log_2 p(x)$$

a.
2. 브라질 vs 중국 축구경기 예시

```
#  vs
>>> 0.99 * -np.log(0.99) + 0.01 * -np.log(0.01)
= 0.056 #    ( )

#  vs
>>> 0.5 * -np.log(0.5) + 0.5 * -np.log(0.5)
= 0.693 #    ()


#        ?
```

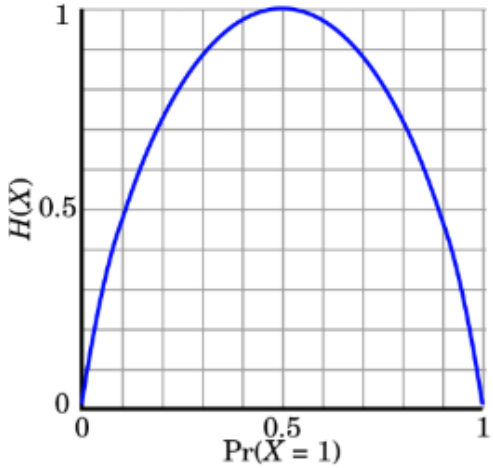
3. P(x) is dependent on the distribution, Non-Uniform 분포의 엔트로피가 Uniform 분포의 엔트로피보다 낮음
   a. If Uniform (8 trials with the same p(x))?

$$H[x] = -8 \times \frac{1}{8}\log_2 \frac{1}{8} = 3 \ bits$$

i.
   b. If Non-uniform?

$$H[x] = -\frac{1}{2}\log_2 \frac{1}{2} - \frac{1}{4}\log_2 \frac{1}{4} - \frac{1}{8}\log_2 \frac{1}{8} - \frac{1}{16}\log_2 \frac{1}{16} - \frac{4}{64}\log_2 \frac{1}{64} = 2 \ bits$$

i.
   c. 동전 던지기처럼 두가지 결과만 있을 경우(베르누이 시행), p가 0.5이면 Entropy(y축)가 최대치에 도달함



d.

e. Entropy for just two choices?

$$H = -p_{correct} \log_2 p_{correct} - (1 - p_{correct}) \log_2 (1 - p_{correct})$$

i.

ii. X = 0, 1인 확률 공간(베르누이 시행)에서 확률값이 다른 3가지 예시 (단, 0 log 0은 0으로 정의)

$$H[X] = -\big[P(X=0)\log P(X=0) + P(X=1)\log P(X=1)\big]$$

- 첫 번째 경우

$$P(X=0) = 0.5$$
$$P(X=1) = 0.5$$
$$H[X] = -(0.5\log 0.5 + 0.5\log 0.5) = 0.69.$$

- 두 번째 경우

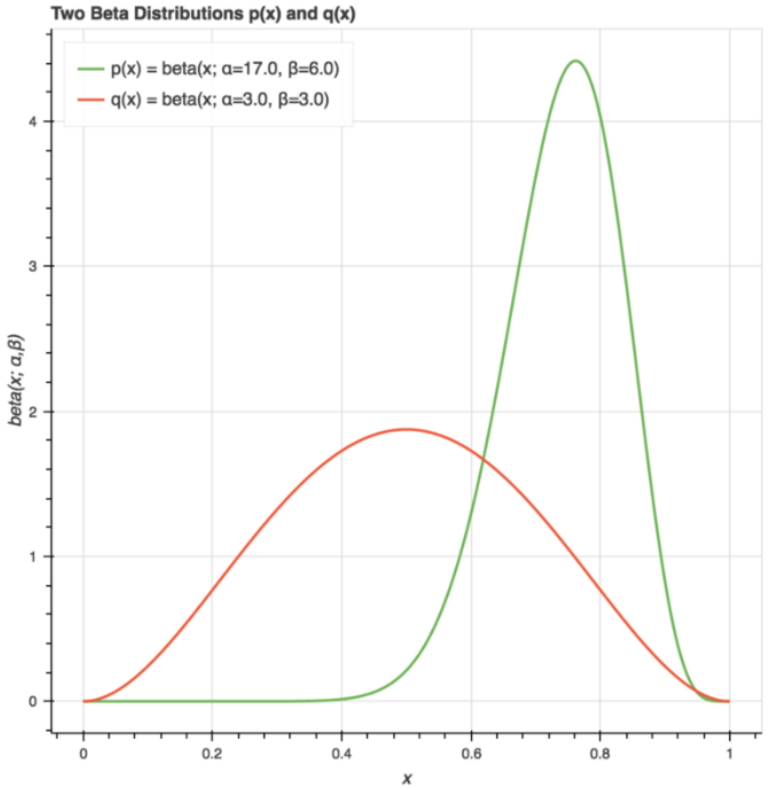
$$P(X=0) = 0.8$$
$$P(X=1) = 0.2$$
$$H[X] = -(0.8\log 0.8 + 0.2\log 0.2) = 0.50.$$

- 세 번째 경우

$$P(X=0) = 1$$
$$P(X=1) = 0$$
$$H[X] = -(1\log 1 + 0\log 0) = 0.$$

iv. KL-Divergence

1. Kullback-Leibler divergence, KLD → 실제 및 예측 분포간의 차이를 표현 (Relative entropy)



a.

2. P(x) 및 Q(x) 가 같을 경우 0
    a. cost function의 기반이 되는 식이지만 거리함수로 이용 불가 (non-symmetric)
    b. P(x): 실제 분포 / Q(x): 모델로 예측 분포

$$D_{KL}\left(P||Q\right) = E_{X\sim P}\left[\log\frac{P(x)}{Q(x)}\right] = E_{X\sim P}\left[\log P(x) - \log Q(x)\right]$$

i.

$$KL(p|q) := -\sum_{i=1}^{N} p_i \log q_i - \left(-\sum_{i=1}^{N} p_i \log p_i\right) = -\sum_{i=1}^{N} p_i \log\left(\frac{q_i}{p_i}\right).$$

ii.

    c. 주요 성질

- $KL(p|q) \neq KL(q|p)$ (non-symmetric).
- $KL(p|q) = 0$ if and only if $p = q$.
- $KL(p|q) \geq 0$.

i.

d. This is convex, and negative log-likelihood 를 통해 최적의 파라메터 θ를 찾을수 있음 (확률적 모수 추정이 가능)
e. 주어진 데이터와 θ에 대해 negative log-likelihood를 최소화 하는 것은 binary cross-entropy를 최소화하는 것과 동치

v. Cross-entropy
1. 실제 P(x)를 알수가 없음. 주어진 데이터셋에서의 p(x)를 이용해 q(x)와 차이를 좁히는 식으로 학습
2. 즉 KLD을 최소화 하는 방향으로 학습 → 크로스엔트로피의 최소화

$$H(P, Q) = -\sum_{x} P(x) \log Q(x)$$

3.
4. p(x)와 q(x)를 서로 교차하여 곱한 값 → 일종의 cost function for classification tasks
5. 크로스엔트로피 계산 예시 (만약 예측과 실제가 다르면 발산, 같으면 0으로 수렴)

$$-P(x) \log Q(x) = -\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \log 0 \\ \log 1 \end{bmatrix} = -(-\infty + 0) = \infty$$

a.

$$-P(x) \log Q(x) = -\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \log 1 \\ \log 0 \end{bmatrix} = -(0 + 0) = 0$$

b.

6. 브라질 vs 아르헨티나 축구경기의 cross entropy
   a. - (P(브라질이 이긴 확률) * log(Q(브라질이 이길 확률)) + P(아르헨티나가 이긴 확률) * log(Q(아르헨티나가 이길 확률)))
7. 만약 브라질(1)이 아르헨티나(0)를 이겼다는 데이터가 있으면?
   a. - (1 * log(Q(브라질이 이길 확률)) + 0 * log(Q(아르헨티나가 이길 확률))) = - log(Q(브라질이 이길 확률))
   b. 즉, log likelihood에 -1 을 곱해준 값 (negative log likelihood)와 동치
8. 논문에 나온 식 다시보기

$$NE = \frac{-\frac{1}{N}\sum_{i=1}^{n}\left(\frac{1+y_i}{2} log(p_i) + \frac{1-y_i}{2} log(1-p_i)\right)}{-(p * log(p) + (1-p) * log(1-p))}$$

a.

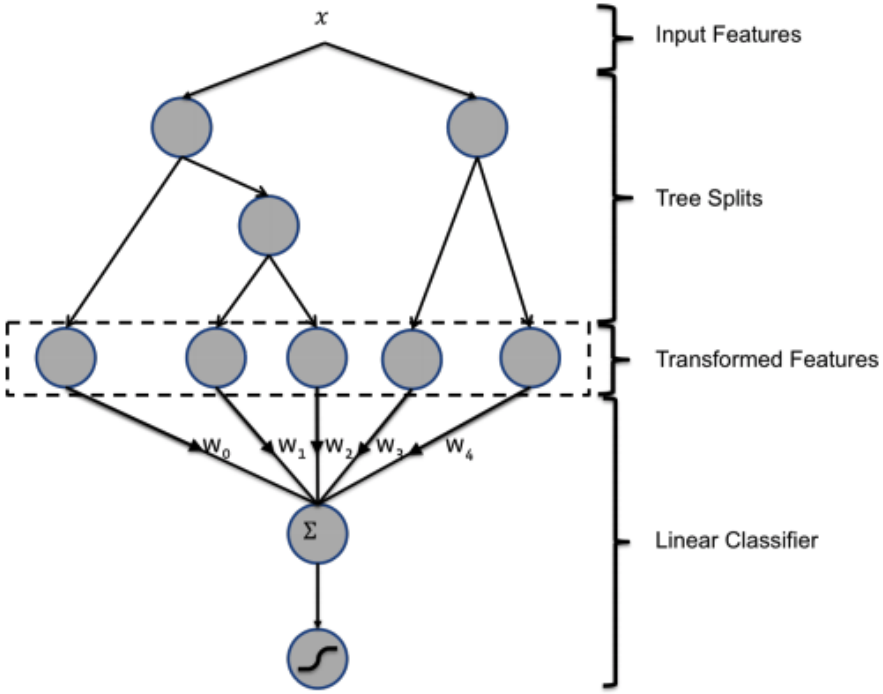
c. RIG (Relative Information Gain)
   i. RIG = 1 -NE
d. Other Metric
   i. Calibration: the ratio of the number of expected clicks to the number of actually observed clicks (1에 가까울수록 좋은 모형)
   ii. AUC(Area under ROC): range from 0.5(base) to 1.0

3. Prediction Model Structure

a. Hybrid model structure

   i. Boosted decision tree → input features transform

ii. The transformed input is treated as a categorical input to a linear classifier.



iii.

b. Decision tree feature transformation
   i. Linear to non-linear → Binning 변환 → 예측도 향상
   ii. Tuple input features
      1. if inputs are categorical → Cartesian product
      2. if continuous → K-d Tree
   iii. The best one was boosted decision trees
      1. In each learning iteration, a new tree is created to model the residual of previous trees → a supervised feature encoding
      2. Source: https://blog.bigml.com/2017/03/14/introduction-to-boosted-trees/

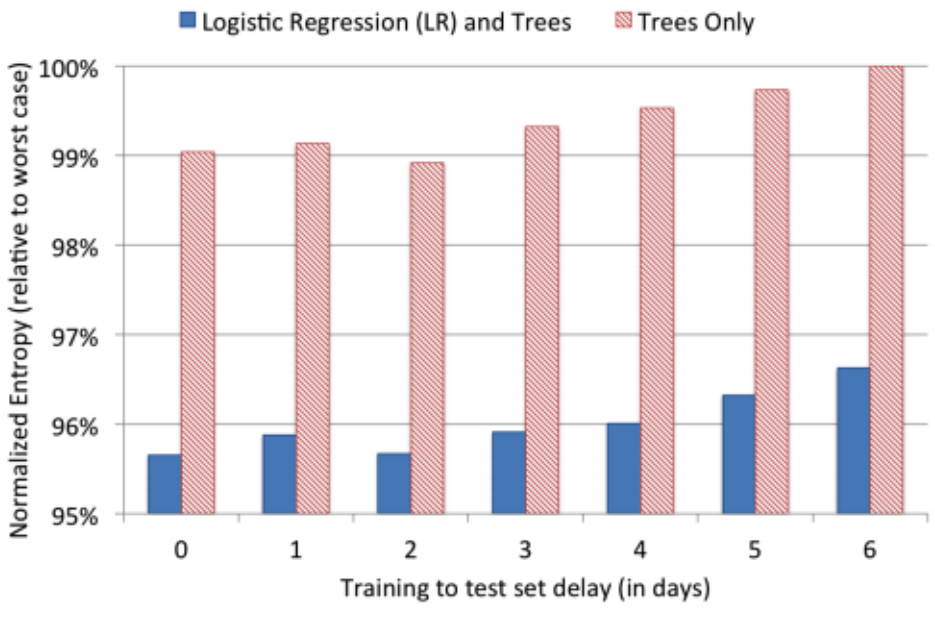
a.

3. Treat each individual tree as categorical feature

    a. 만약 instance가 2번째, 4번째에 해당되면 1-of-K 코딩으로 [0,1,0,1,0] 으로 코딩

4. Experiment result
    a. LR with plain features vs LR with transformation by boosting decision tress
    b. feature transformation help decrease NE by 3.4%

Model Structure	NE (relative to Trees only)
LR + Trees	96.58%
LR only	99.43%
Trees only	100% (reference)



c.

c. Data freshness
   i. 시간의 흐림에 따라 데이터 분포가 변화하는 부분을 고려하여, 데이터의 freshness 를 측정하고 평가함
   ii. 그 결과, 일주일 단위(training weekly) 학습보다 하루 단위 학습(training daily)의 경우 NE가 1% 감소시키는 것으로 나타남
      1. Boosted decision tree → 하루 단위(daily)로 학습
      2. Linear Classifier → 준실시간으로 학습 (online learning)
d. Online linear classifier
   i. data freshness 확보를 위해 준실시간으로 학습 가능한 infrastructure 구축(4장에서 다시 소개) & 테스트
   ii. learning rate 탐색후 SGD-based, BOPR model 이용해 accuracy 측정
   iii. Best Learning rate was Per-coordinate Learning Rate
   iv. *Note: 1~3 번은 변수별 learning rate이 다름*

1. Per-coordinate learning rate: The learning rate for feature i at iteration t is set to

$$\eta_{t,i} = \frac{\alpha}{\beta + \sqrt{\sum_{j=1}^{t} \nabla_{j,i}^2}}.$$

   α, β are two tunable parameters (proposed in [8]).

2. Per-weight square root learning rate:

$$\eta_{t,i} = \frac{\alpha}{\sqrt{n_{t,i}}},$$

   where $n_{t,i}$ is the total training instances with feature i till iteration t.

3. Per-weight learning rate:

$$\eta_{t,i} = \frac{\alpha}{n_{t,i}}.$$

4. Global learning rate:

$$\eta_{t,i} = \frac{\alpha}{\sqrt{t}}.$$

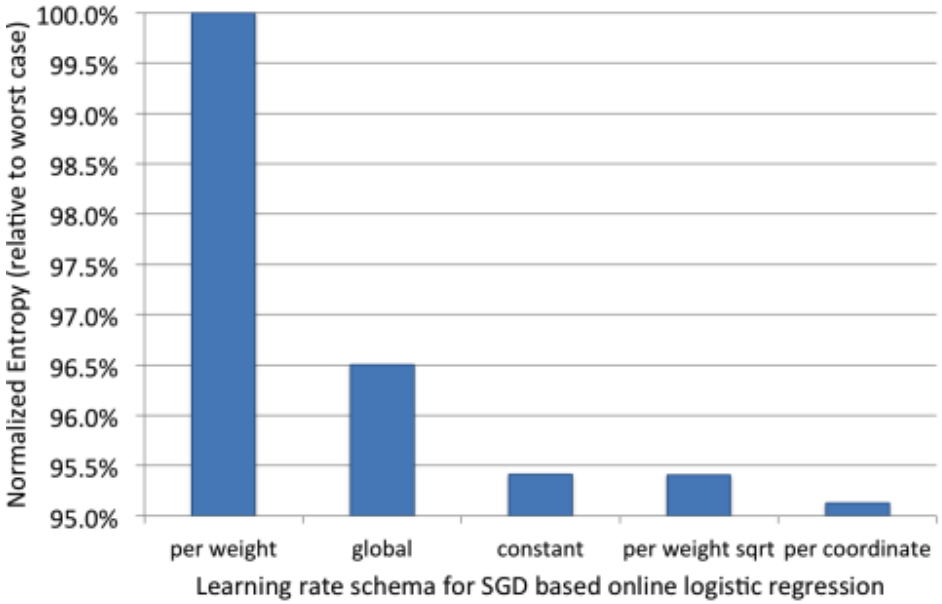
5. Constant learning rate:

$$\eta_{t,i} = \alpha.$$

v. The X-axis corresponds to different learning rate scheme. The normalized entropy is shown with the y-axis.

Table 2: Learning rate parameter

Learning rate schema	Parameters
Per-coordinate	$\alpha = 0.1, \beta = 1.0$
Per-weight square root	$\alpha = 0.01$
Per-weight	$\alpha = 0.01$
Global	$\alpha = 0.01$
Constant	$\alpha = 0.0005$



vi.

4. Online Data Joiner
   a. 실시간으로 데이터를 학습하고 실험하기 위한 시스템 → Online joiner
      i. it joins label(click/no-click) to training inputs stream (ad impression) in an online manner

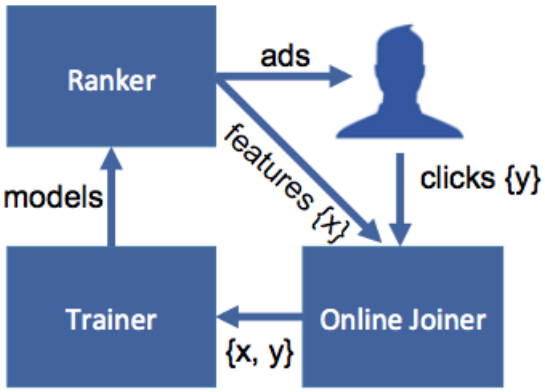


Figure 4: Online Learning Data/Model Flows.

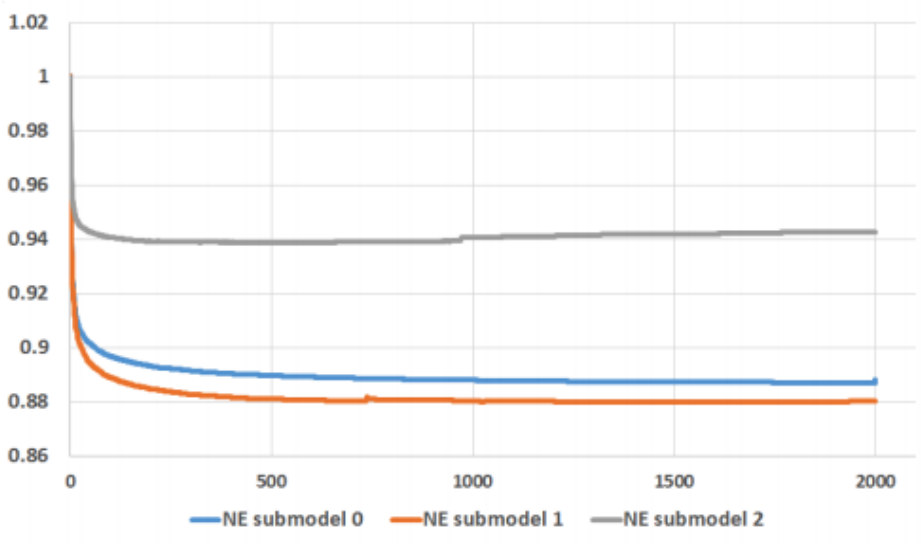
ii.

b. Recency and click coverage* → online joiner system 구축시 필수 고려 요소 / *조인후 not-null의 coverage

   i. Click은 명확히 판단 가능하나 No-click 은 모호함 → 기준 필요
   ii. 특정 시간이 지난후에 아무 반응이 없을 경우 No-click으로 정의
   iii. 기준 시간을 정의할 때 메모리 성능 고려 if too long → impression buffering increases, if too short 클릭 정보 lost
   iv. insufficient coverage → biased
   v. Designed to perform a distributed stream-to-stream join on ad impressions
   vi. A tight closed loop for the machine learning models
   vii. Needs to build protection mechanisms against anomalies → It will lower the accuracy of the model

5. Containing Memory and Latency
   a. Number of boosting trees → tree 수에 따른 예측 정확도 차이 파악 study for computation costs

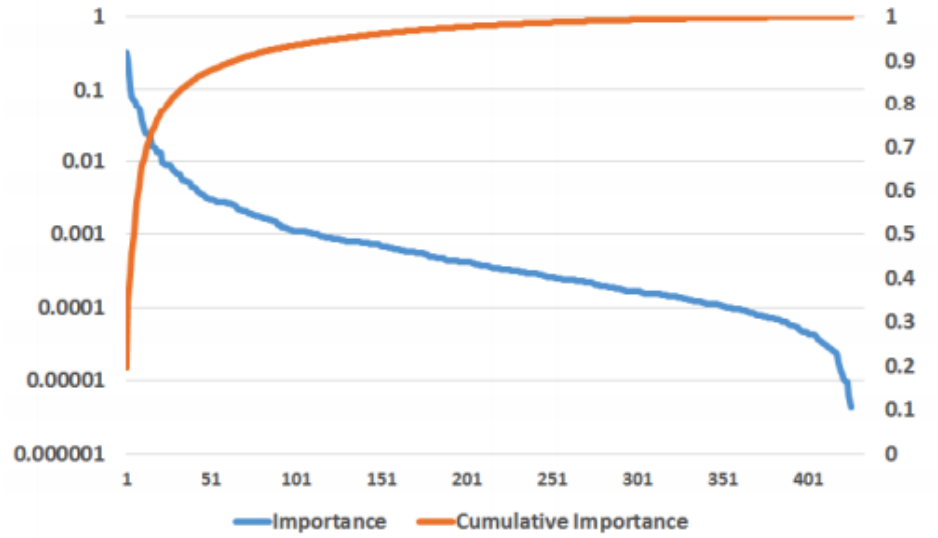
      i. X-axis → the number of boosting tree / Y-axis → the normalized entropy
      ii. 500개까지 NE 감소, 100개부터 거의 감소하지 않음.

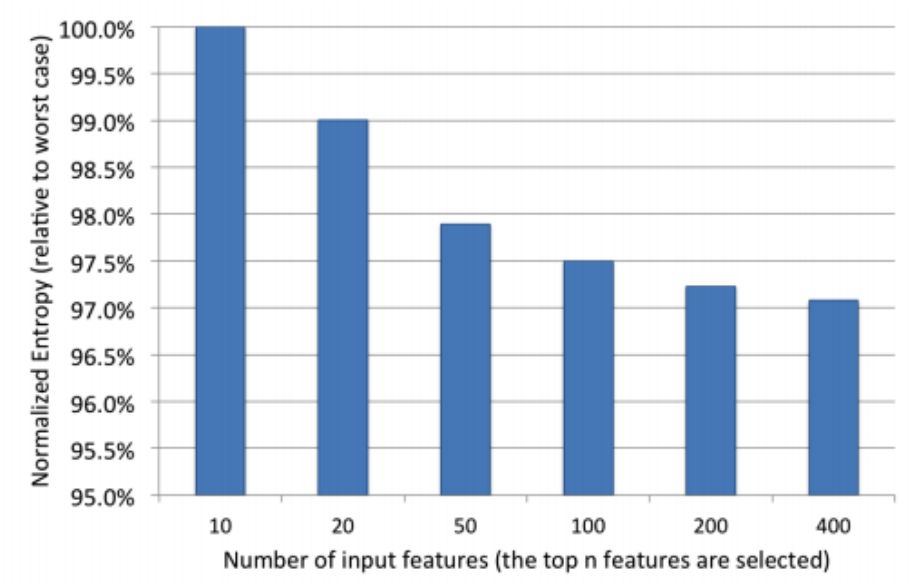


iii.

b. Boosting feature importance → 변수의 수는 정확도 및 Computation costs에 영향
   i. X-axis → the number of features / Y-axis → left: feature importance in log scale, right: cumulative feature importance
   ii. *10개의 변수로 누적50% 도달*, 10, 20, 50, 100, 200, 400으로 구분해 NE 측정

iii.





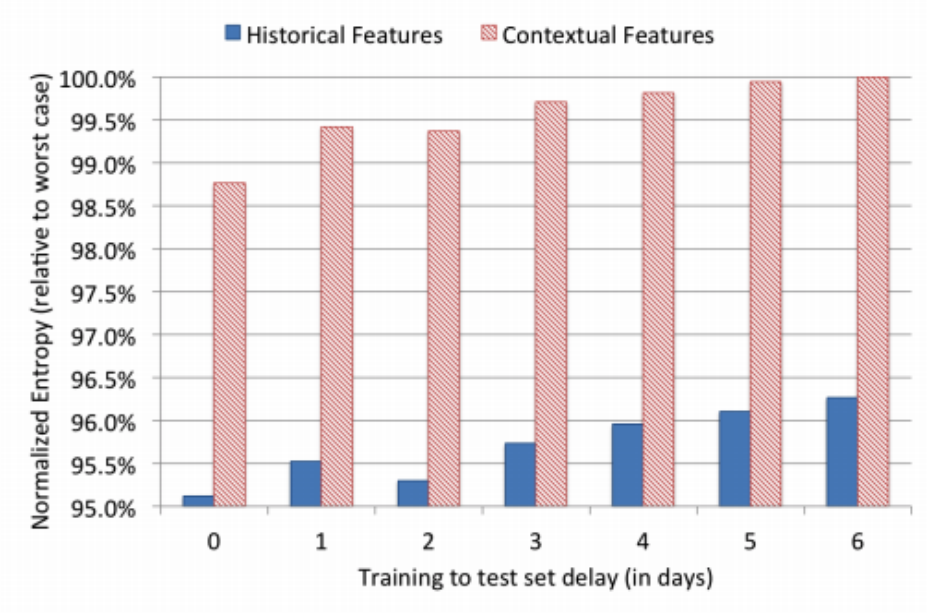
c. Historical features
   i. Contextual features: 현재 사용중인 디바이스, 현재 머물고 있는 페이지, 시간대 및 요일 등
   ii. Historical features: 과거의 방문기록, 누적클릭수, CTR 등
   iii. 위 2 타입별의 상대적 성과 측정 with the percentage of top k-important features after sorting by all features by importance
   iv. historical feature occupying roughly 75% of the features in this dataset while conceptual features were only 2

v. X-axis: 변수 개수, Y-axis: 상위 K 개 중요변수중 historical feature 개수

vi.

Type of features	NE (relative to Contextual)
All	95.65%
Historical	96.32%
Contextual	100% (reference)

vii.

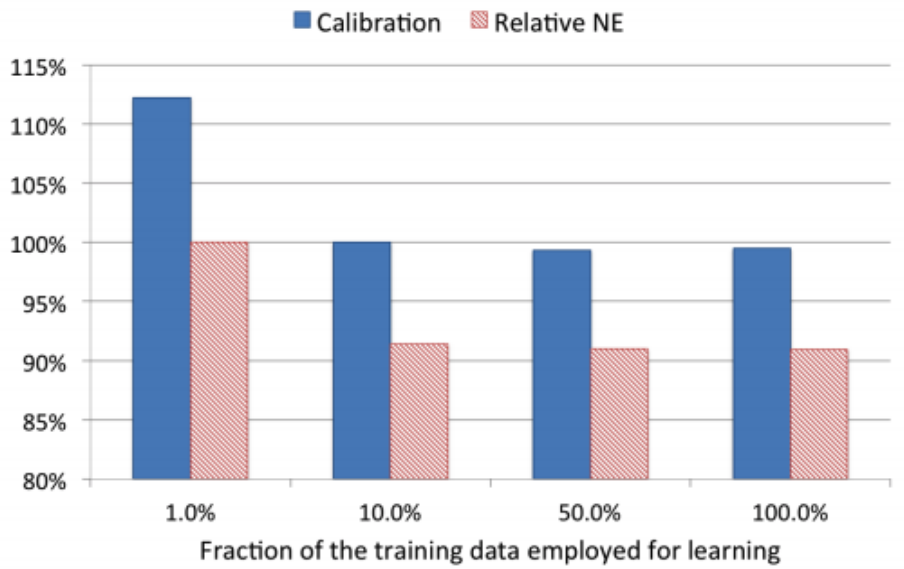


viii.

ix. But contextual features are very important to handle the cold start problem.

x. We can see that the model with contextual features relies more heavily on data freshness while historical data is stable

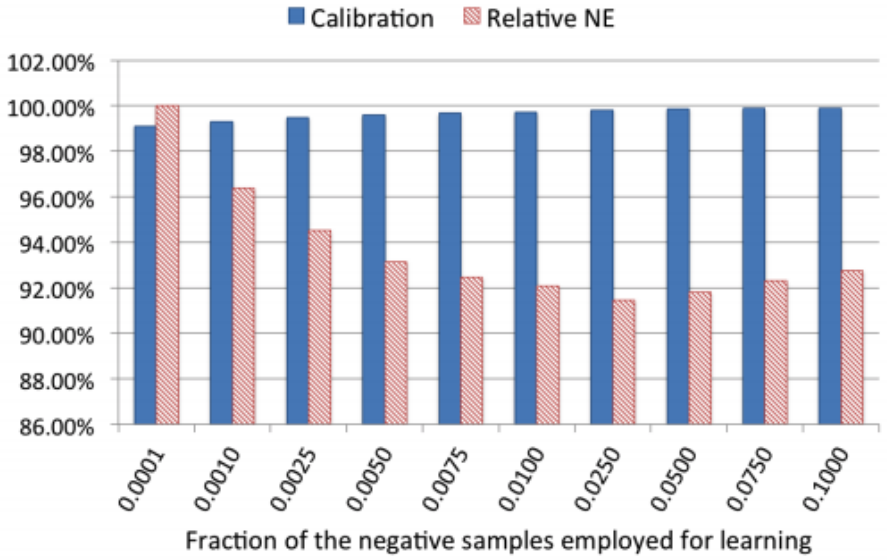
6. Coping with Massive Training Data
   a. Dataset is massive
   b. 1) Uniform subsampling rate: {0.001, 0.01, 0.1, 0.5, 1}. As result, 10% is not so different from 100%



i.

c. 2) Negative down sampling for solving class imbalance problem

i. rate: {0.1, 0.01, 0.001, 0.0001}: the best one was 0.025



ii.

iii. https://medium.com/bluekiri/dealing-with-highly-imbalanced-classes-7e36330250bc

- **Undersampling** (*balanced*): all the observations from the minority class are kept and sampling without replacement is performed in the majority class where the number of observations sampled is equal to the sample size of the minority class.

- **Upsampling** (*balanced*): decide how many times the sample size of the minority class wants to be used and perform sampling with replacement in the minority class and sampling without replacement in the majority class. Be careful with this strategy because it has the risk to trigger overfitting due to the repetition of the same observations in the minority class.

- **Negative downsampling** (*imbalanced*): different samples sizes are used in this procedure. In all these samples all the observations from the minority class are kept and we take different number of observations from the majority class by performing sampling without replacement.

7. Discussion
    a. Data freshness matters. It is worth retraining at least daily
    b. Transforming real-valued input features with boosted decision trees significantly increases the prediction accuracy of probabilistic linear classifiers.
    c. Best online learning method: per-coordinate learning rate
    d. How to keep memory and latency
        i. the number of boosted decision trees
        ii. features by feature selection by means of feature importance.
        iii. historical features in combination with context features
        iv. subsampling
8. 생각해볼 문제들 in action
    a. 모델의 하이브리드 효과가 존재하는 것 같다. (ensemble)
        i. 복수의 모델을 써야 할 상황이 생길 것 → 시스템 및 모델 성능 간 trade-off 고려
        ii. 여러 모델을 테스트하고 최적의 조합 도출 필요
    b. Feature selection, Parameter tuning 을 통한 모델 및 시스템 성능 개선 필요
        i. 변수 생성, 검증이 중요한 단계일 것
        ii. 유저 변수, 아이템 변수
    c. Data freshness, Learning rate, sampling 은 부차적인 요소
        i. 모델 성능이 소폭 향상되는데 도움을 줄 것이나, 적절한 변수/모델 선정에 비해 영향력이 낮음 → 우선순위: 변수와 모델
        ii. But Data freshness 는 변수의 가중치 요소로 활용 가능할 것
9. Next Step
    a. 신규 변수 생성 및 검증
    b. 모델 생성/조합 및 테스트
        i. online metric
        ii. offline metric 개발
    c. 유형별 테스트 결과 정리

d. 논문 리딩