

Customer Lifetime Value Prediction Using Embeddings

0. ABSTRACT

- 2017년 7월 발표, ASOS.com에서 배포된 CLTV 예측 시스템을 설명
- 매번 바뀌는 상품 목록, feature 수동 생성의 어려움을 해결하고 개선하는 embedding 생성에 대해 제안할 것

1. INTRODUCTION

- ASOS는 global e-commerce company, 패션과 뷰티에 특화되었음
- 무료 배송 및 반품이 BM의 필수요소이지만, 이 때문에 소비자가 쉽게 부정적인 lifetime value를 가지게 됨
 - 이 부분에서 CLTV 측정이 온라인 의류 소매업에서 중요해짐
- ASOS에서는 CLTV를 소비자 개인이 1년간 구매한 총액으로 정의함
- 목적이 되는 Business Metric은 총 3가지임
 - 평균 구매빈도 예측 → 빠르게 '될성부른' 유저를 찾아 고가치 유저로 육성하는 근거로 활용
 - 평균 구매량 예측 → 빠르게 '될성부른' 유저를 찾아 고가치 유저로 육성하는 근거로 활용
 - 이탈률 예측 → 이탈 가능성이 높은 유저를 찾아, 리텐션 캠페인에 들어가는 비용을 조절
- 사람이 직접 만드는 feature는 human bottleneck 존재
 - 이는 데이터의 full richness를 활용하지 못하는 이유가 됨
 - 자동으로 학습하는 feature를 조합하여, 도메인 지식과 소비자의 멀티 패턴을 모두 활용하는 모델을 생성
- 학습용 데이터는 지난 2개년의 feature와 label을 사용

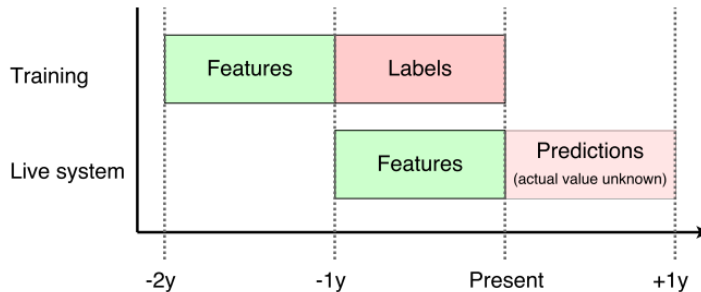
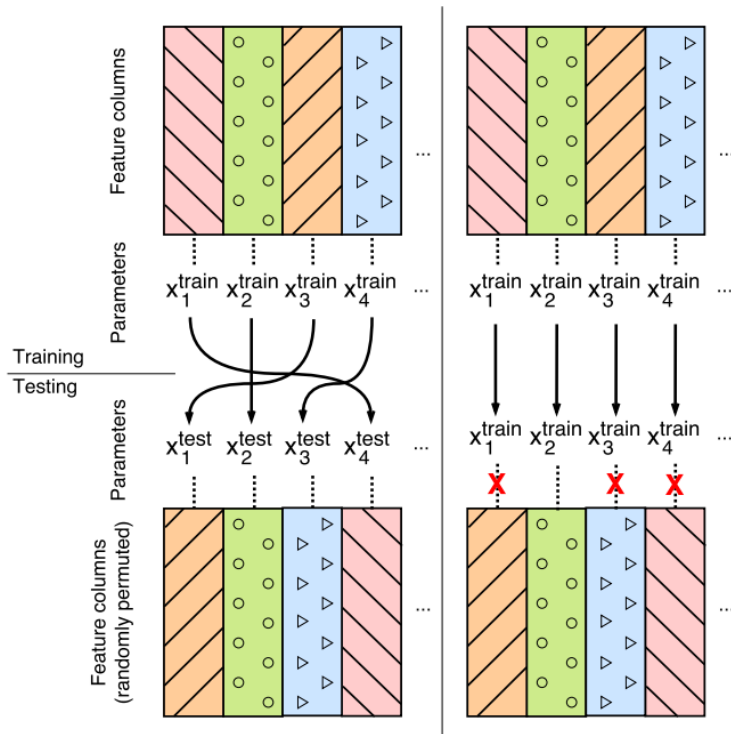


Figure 1: Training and prediction time-scales for CLTV. The model is retrained every day using customer data from the past two years. Labels are the net customer spend over the previous year. Model parameters are learned in the training period and used to predict CLTV from new features in the live system.

- 만들어진 feature의 적용이 어려운 부분이었음
 - 차원축소된 feature는 따로 labeling된 것이 아니므로, 매 period마다 새로 weight를 학습시킨다면 이전에 축소되었던 순서대로 feature가 생성되지 않음
 - feature의 순서가 다르면 학습이 제대로 되지 않음
 - warm start for the test period embedding으로 해결
 - 학습 시기에 존재했던 유저 : 학습 시기에 사용했던 weight로 initialisation
 - 학습 시기에 존재하지 않았던 신규 유저 : uniform random value로 initialisation 하되, 학습시기의 weight보다는 그 절댓값이 작은 값이어야 함
 - 이렇게하면 weight의 구조에 큰 차이가 발생하지 않으므로, feature의 순서도 어느 정도 유지된다고 보는 것 같음



2. RELATED WORK

2.1 Distribution Fitting Approaches

- CLTV를 추정하는 데에 사용되었던 모수적 방법론을 소개함
 - NBD (negative-binomial distributed)
 - RFM (Recency-Frequency-Monetary Value)
 - 이 방법론들은 sparse feature에 취약하다는 단점이 존재, 머신러닝의 필요성이 부각됨

2.2 Machine Learning Methods

- 조사해본 바로는, (당시까지) 뉴럴넷이 CLTV 문제에 효과적으로 적용된 사례는 없음
- 다른 도메인에서 비슷한 문제에 CNN과 autoencoder를 사용한 것까지는 확인

2.3 Neural Embeddings

- 가장 널리 알려진 임베딩 모델로 SGNS가 있음 (SkipGram with Negative Sampling)
 - item-based CF에 사용됨
 - 핵심은 sequence data를 활용했다는 점

3. CUSTOMER LIFETIME VALUE MODEL

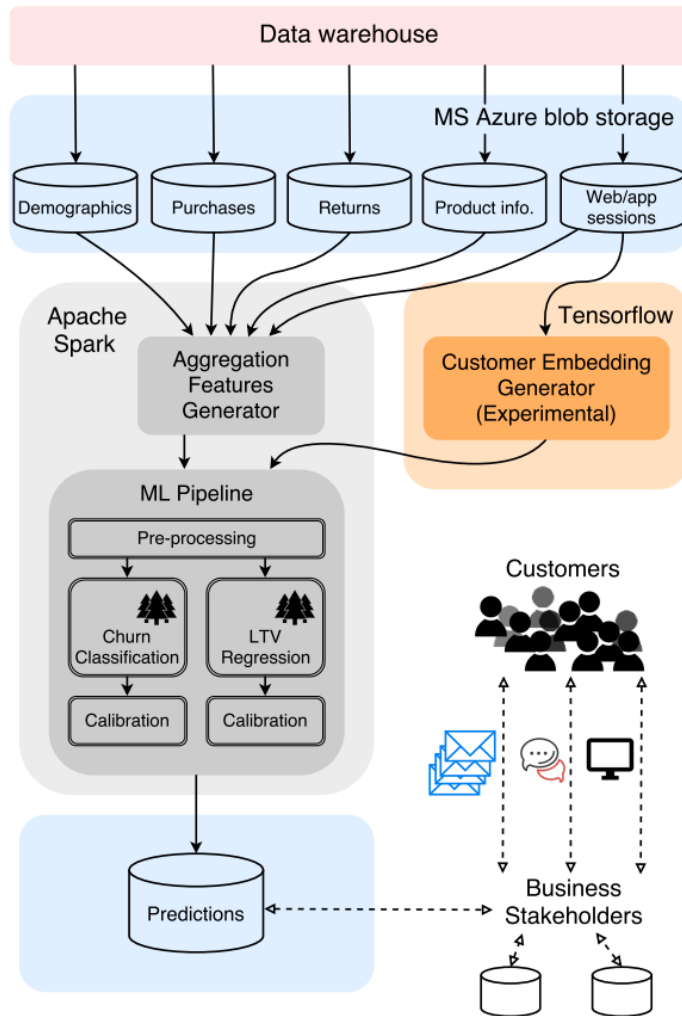
- ASOS의 CLTV 모델은 풍부한 데이터를 바탕으로 이후 12개월 동안의 유저의 총 소비량을 예측함
- 가장 문제가 되는 점은 target value의 분포 형태임
 - 대부분의 유저의 CLTV는 0이고, 0 이상인 유저들도 그 규모가 여러 층으로 나누어짐
 - 이 문제를 해결하기 위해, RF regressor로 CLTV percentiles를 모델링함 (금액이라는 real value를 모델이 예측하기 쉬운 형태로 변환)
 - 예측된 percentile 값은 real value와 매칭됨

3.1 Features

- 모델에 사용된 Feature의 종류는 크게 4가지임
 - 사용자의 인구통계 정보
 - 구매이력
 - 반품이력
 - 웹과 앱에서의 세션 로그

- RF의 feature importance로 순위를 매겨본 결과, 일반적 예상에서 벗어나는 중요도를 보인 feature가 존재했음
 - 주문일자의 표준편차 (주문하는 term이 얼마나 일정한가?)
 - 국적
 - 새로운 아이템에 대한 반응 → 패션업계에서의 고가치 유저는 새로운 아이템에 반응함

3.2 Architecture



3.3 Training and Evaluation Process

- 예측 시점 기준, 지난 해의 데이터를 사용하여 매일 모델을 재학습
 - 매일 집계되는 유저별 feature와 embedding, target label을 생성
 - 스코어로는 AUC를 사용

3.4 Calibration

- 모델의 output이 realised value와 같지 않으므로, 그것을 보정한다는 의미의 Calibration
 - 이탈 예측 모델
 - RF의 이탈예측치(estimates)만을 사용해서 이탈여부를 예측하는 LR 모델을 생성
 - LR의 output을 calibrated probability로 해석
 - CLTV 예측 모델
 - RF의 Regression estimates는 RMSE를 minimize하는 방식으로 얻어지는데, 이것이 실제 realised CLTV의 분포와 같으리라는 보장이 없음
 - 이 문제 해결을 위해, 일단 CLTV percentile을 예측하게 한 뒤에 이를 실제 monetary value와 매핑함
 - 매핑에는 decision tree를 학습시켜 사용함
- Calibration의 이점
 - 아웃라이어에 대해 상대적으로 robust한 모델이 생성됨

- 실제 값과 더 정확하게 일치하는 예측치를 얻음

3.5 Results

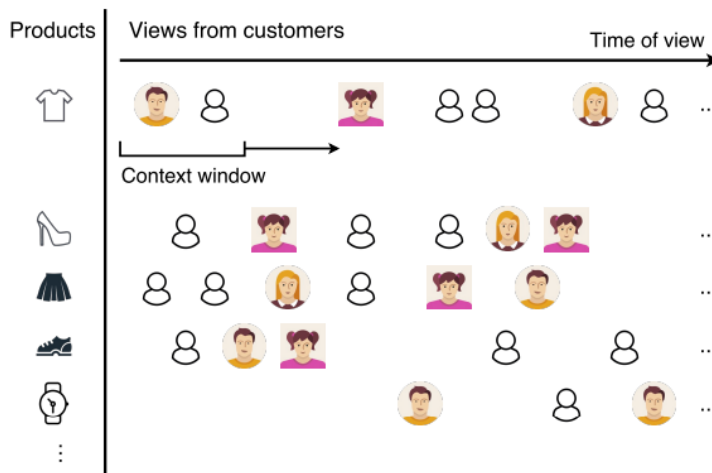
- 10-fold CV 진행
- Spearman rank-order correlation coefficients로 CLTV 예측성능 측정
 - 0.56 (전체 유저 대상), 0.46(실구매 유저 대상)
 - 예측 분포의 범위와 밀도가 실제와 유사함
- AUC로 이탈예측 성능측정
 - 0.798의 스코어

4. IMPROVING THE CLTV MODEL WITH FEATURE LEARNING

- 2가지의 접근방법
 - 유저의 상품 view를 사용하여 embedding을 생성
 - LR과 DNN을 결합한 하이브리드 모델을 학습

4.1 Embedding Customers using Browsing Sessions

- 각 상품별로, 해당 상품을 조회한 유저의 sequence를 마치 자연어에서의 단어의 sequence처럼 사용
 - 유저와 상품의 interaction에 기초한 상품 embedding을 활용할 수도 있으나, 상품의 수명이 짧은 패션 업계에서는 적절하지 않음
 - 직관적으로, 고가치 유저는 고가의 상품, 덜 유명하거나 최저가가 아닌 상품을 찾고 저가치 유저는 세일 상품이나 최저가 상품을 찾음
 - 이 정보는 수동으로 만들어지는 feature로 반영하기 어렵다



- SGNS(SkipGram with Negative Sampling)를 적용하기 위해 필수적이었던 3개의 key design
 - context를 어떻게 정의할 것인가?
 - length = 11 (경험적으로 찾음)
 - context 안에서 유저의 pair를 어떻게 만들어낼 것인가?
 - 11명의 유저가 있으면, 가장 중심에 있는 유저를 두고 10개의 쌍이 만들어짐
 - negative sample은 어떻게 만들어낼 것인가?
 - k개의 샘플을 뽑아 가중치 업데이트에 반영
- hidden layer의 차원을 늘릴수록 모델 성능도 개선되는가?
 - 일정 수준까지는 증가하나, 특정 값 이후로는 의미 없음
- 4.1.1 Embeddings for the Live System
 - 새로운 유저의 embedding은 기존 유저의 embedding의 linear combination으로 생성한다는 점이 핵심
 - 유저의 경우, 과거부터 현재까지 n년간 존재하는 경우가 많지만 상품은 그렇지 않으므로 유저 embedding을 생성

4.2 Embeddings of Handcrafted Features

- 기존의 RF 모델을 DNN으로 개선하려 시도
- 결과적으로, 상향된 퍼포먼스로 얻는 이익보다 모델 훈련에 들어가는 비용이 더 컸음

5. DISCUSSION AND CONCLUSIONS

- 위 내용 요약과 같음

