KIPS Trans. Softw. and Data Eng.
Vol.12, No.6 pp.259~266
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2023.12.6.259

Quantitative Estimation Method for ML Model Performance Change, Due to Concept Drift

Soon-Hong An† · Hoon-Suk Lee†† · Seung-Hoon Kim†††

ABSTRACT

It is very difficult to measure the performance of the machine learning model in the business service stage. Therefore, managing the performance of the model through the operational department is not done effectively. Academically, various studies have been conducted on the concept drift detection method to determine whether the model status is appropriate. The operational department wants to know quantitatively the performance of the operating model, but concept drift can only detect the state of the model in relation to the data, it cannot estimate the quantitative performance of the model. In this study, we propose a performance prediction model (PPM) that quantitatively estimates precision through the statistics of concept drift. The proposed model induces artificial drift in the sampling data extracted from the training data, measures the precision of the sampling data, creates a dataset of drift and precision, and learns it. Then, the difference between the actual precision and the predicted precision is compared through the test data to correct the error of the performance prediction model. The proposed PPM was applied to two models, a loan underwriting model and a credit card fraud detection model that can be used in real business. It was confirmed that the precision was effectively predicted.

Keywords : Concept Drift, Data Drift, Covariate Shift, Kolmogorov-Smirnov Test

Concept Drift에 의한 ML 모델 성능 변화의 정량적 추정 방법

안 순 홍† · 이 훈 석†† · 김 승 훈†††

요 약

기계학습을 통해 학습된 모델은 업무 활용 시 그 성능을 실측하기 매우 어렵다. 때문에 운영 부서에서는 모델의 성능을 효과적으로 관리하지 못한다. 이로 인해 모델의 상태를 판단하기 위한 Concept drift 탐지 방법이 다양하게 연구되고 있다. 운영 부서에서는 운영 중인 모델의 성능을 정량적으로 관리하려고 한다. 그러나 Concept drift는 모델 상태를 데이터 관계적으로 판단 할 뿐, 모델의 정량적 성능 수치를 추정하지는 못한다. 본 연구에서는 Concept drift의 통계량을 통해 정량적으로 precision 값을 추정하는 성능 예측 모델(PPM, Performance prediction model)을 제안한다. 제안 모델의 Algorithm 1에서는, 학습데이터에서 복원 추출한 샘플링 데이터에 인위적인 drift를 유도하고 이때의 precision을 측정하여 drift와 precision의 데이터 셋을 만들어 학습한다. Algorithm 2에서는 테스트 데이터를 통해 실제 precision과 예측 precision의 차이를 측정하여 성능 예측 모델의 오차를 보정 한다. 현실 비즈니스에서 사용될 수 있는 대출 심사 모델과 신용카드 오사용 탐지 모델에 PPM을 적용하여 성능 예측의 유효성을 확인했다.

키워드 : 성능 예측 모델, 예측변화, 공변량, KS테스트

1. 서 론

기계학습 및 머신러닝이 일반화 되면서 수많은 모델이 생성되고 비즈니스 업무에 활용되고 있다. 그러나 이러한 모델은 시간의 변화, 외부의 다양한 변수 등에 의해 최초 생성된 시점에 비해 모델의 성능은 점점 떨어지게 된다. 이러한 모델의 성능 쇠퇴를 Model drift, Model decay, Model staleness[1]등으로 일컫는다. Model drift는 Concept drift와 Data drift로 구분 할 수 있다. 모델에 대한 개념은 $p(X, Y) = p(Y|X)p(X)$ 로 표현 할 수 있으며, 예측하려고 하는 $p(Y|X)$의 Y, X의 상관관계가 바뀌는 경우를 Concept drift[2], Y, X의 상관관계는 변함 없이 $p(X)$의 통계적 속성이 변하는 경우를 Data drift[2]라고 한다.

1.1 Concept Drift

Concept drift는 다른 용어로 Class drift, Real concept drift[3]로도 불리며, $p(Y|X)$의 변화를 나타낸다. 이는 $p(X)$의 변경 여부에 관계없이 발생[4] 되는 것을 의미한다.

Fig. 1은 (b)Concept drift(Real concept drift)와 (c) Data drift(Virtual Concept drift)를 비교한다. (a)Original 데이터에서 Concept drift는 $p(Y|X)$의 변경으로 클래스 경계가 변경되어 이전에 만들어진 의사 결정 모델은 의미가 없어진 것을 보여 준다.

† 준 회 원 : 아시아나IDT AI빅데이터연구소 수석연구원
†† 정 회 원 : 아시아나IDT AI빅데이터연구소 수석연구원
††† 정 회 원 : 단국대학교 컴퓨터공학과 교수
Manuscript Received : September 30, 2022
First Revision : January 13, 2023
Accepted : January 25, 2023
* Corresponding Author : Soon-Hong An(kingmir@dankook.ac.kr)

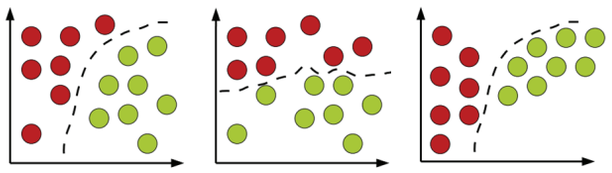

Fig. 1. Types of Drifts: Circles Represent Instances, Different Colors Represent Different Classes

1.2 Data Drift

데이터 분포 변화로 인해 현재 모델에 영향을 미치는 것을 Data drift라고 하며, 다른 용어로는, Virtual drift, Virtual concept drift[3]라고 한다. 이는, $p(X)$는 변경 되지만 $p(Y|X)$는 변경 되지 않는 것으로 데이터 분포의 특성은 변경되지만, 모델의 성능은 변경되지 않는 것을 의미한다.

1.3 Concept Drift의 문제점

일반적으로 측정된 성능 값은 해당 모델을 대변하는 성능 지표가 된다. 이렇게 구축된 모델은 live data stream (online data)을 통해 비즈니스 환경에 활용한다. 그러나, 초기에 측정된 모델의 성능이 지속적으로 보장되는지 실측을 통해서는 알 수 없다.

Regression을 통해 Time-series 데이터로 모델을 만들고 예측을 할 경우, 실측 Y값은 알 수 있으나 역시 업무에 따라 그 시간 간격이 크게 발생 할 수 있다. 그리고, Classification의 경우, 예측 class가 정확한지 실측하기 위해서는 별도의 labeling 작업을 해야 하는 어려움이 존재한다.

결과적으로 학습을 통해 구축된 모델은 활용 단계에서 실시간으로 성능을 측정하는 것이 쉽지않다. 또한 비즈니스 환경의 IT부서에서 각각의 분석 모델에 대한 성능의 threshold를 운영, 관리, 유지보수 하는 것은 현실적으로 매우 어렵다.

이러한 비즈니스 환경에서 분류, 분석 모델 활용시 성능 저하를 인지하지 못하고 지속적으로 사용 할 경우 기업의 의사결정 및 경영상의 문제를 초래 할 뿐 아니라 막대한 경제적 손실로 이어질 수 있다.

위와 같은 문제를 해결하기 위해 concept drift의 통계량을 측정하여 모델의 성능 상태를 유추하는 연구들이 이루어져 왔다. 하지만, 대부분의 연구가 Ground Truth가 입력되어야 하거나 모델이 운영 중인 경우 적당한 Heuristic Threshold를 필요로 하고있어 실제 비즈니스 환경에서 적용하기에는 어려움이 존재한다.

만약, concept drift로 발생되는 성능 저하의 정량적 성능 값(precision, recall, accuracy 등)을 직접적으로 추정할 수 있다면, 실제 비즈니스 환경에서는 모델의 성능 관리에 매우 유용하게 사용될 것이다.

이를 위해, 이전부터 detecting dataset shift, Dimensionality Reduction, Statistical Hypothesis Testing, Obtaining Most Anomalous Samples, Determining the Malignancy of a Shift 등에 대한 다양한 연구가 진행 되었다.[5]

하지만, 이렇게 수많은 연구에서도 Concept drift 발생시 모델 성능을 정량적으로 예측하는 연구는 많지 않았다.

1.4 본 연구의 목적

본 연구에서는 Concept drift 측정값 (통계량)을 통해 직접적으로 모델의 성능을 정량적으로 예측하는 성능 예측 모델(PPM, Performance prediction model)을 제안한다. 이를 통해 운영 중인 모델의 성능을 추정하고 모델의 성능 저하가 추정되면 재학습을 통해 모델 성능의 항상성을 지속적으로 유지 할 수 있다.

2. 관련 연구

Drift가 발생되면 최초 생성된 모델은 의도와 다른 결과를 제공하고, 이는 비즈니스 의사결정에 부정적인 영향을 미치게 된다. 이러한 문제점을 해결하고 Drift를 극복하기 위해 다양한 연구가 진행 되고 있다.

Drift를 감지하는 방법으로 전통적인 데이터 분포에 기반한 방법과 복수의 알고리즘을 조합한 ensemble 방법[6][7]이 주를 이루고 있다.

2.1 데이터 분포에 기반한 방법

1) DDM(Drift Detection Method)[8] : 오류율(pi)과 표준편차(si)를 모니터링하여 Drift 감지를 한다.

2) EDDM(Early Drift Detection Method)[9] : DDM과 유사하나 오류율(pi)가 아닌 연속오류 사이의 거리를 분석하여 Drift 감지를 한다.

3) ADWIN(Adaptive Windowing)[10] : 이전 데이터와 최신 데이터를 동적으로 분할된 두 개의 인스턴스(w) sliding window로 나눠 w의 증감으로 Drift 감지를 한다.

4) SeqDrift(Sequential Drift)[11] : ADWIN을 기반하며, 고정된 window를 통해 sampling 데이터를 관리하고 Bernstein bound[12]를 활용하여 표본 평균의 분산(분포) 계산으로 Drift 감지를 한다.

5) SEED(state-of-the-art drift detection)[13] : ADWIN을 기반하며, 두 개의 window를 비교하여 window의 평균 임계값 및 Bonferroni 보정, Hoeffding 부등식 계산을 통해 Drift 감지를 한다.

6) STEPD(statistical test of equal proportions to detect)[14] : 두 window의 연속성 보정 및 동일한 비율의 통계적 검증을 통해 Drift 감지를 한다.

7) WSTD(Wilcoxon Rank Sum Test Drift Detector)[15] : STEPD을 기반하며, STEPD과 다르게 동일한 비율을 사용하는 대신 Wilcoxon[16] 순위 합의 통계 및 추가 변수로 window 크기 제한을 통해 Drift 감지를 한다.

8) FTDD(Fisher Test Drift Detector)[17] : STEPD을 기반하며, Fisher's Exact test [18]를 통해 소형, 불균형 데이터 샘플을 동일한 비율의 통계적 검증으로 Drift 감지를 한다. 파생 알고리즘으로 Fisher's Exact test를 hybrid 방식

으로 반영한 FPDD(Fisher Programs Drift Detector)[19]와 FSDD(Fisher Square Drift Detector)[19]이 있다.

2.2 복수 알고리즘 조합 방법

1) DWM(Dynamic weighted majority)[20] : Weighted Majority Algorithm(WMA)[21]을 확장한 가중 ensemble 알고리즘이며, 전체 성능에 따라 분류기를 추가 또는 제거한다. Drift를 감지하는 것에 초점을 맞추기보다 Drift에 적응을 하는 것을 목표로 한다.

2) Bagging [22] 및 Boosting [23] : 알고리즘의 정확도를 위해 사용하는 방법으로 Bagging은 반복된 training set에서 다시 sampling하여 training을 위해 무작위로 생성된 서로 다른 bootstrap sample을 사용한다. 그리고, Boosting은 training Data에 대해 서로 다른 분포를 사용하여 여러 학습자를 training시키고 이전 예측에 따라 각 분류자에게 주어진 다양성의 양을 변화시킨다. 이러한 값을 집계하여 ensemble로 결합한다.

3) OzaBag(Oza and Russell's Online Bagging), OzaBoost (Oza and Russell's Online Boosting)[24] : Poisson 분포를 사용하여 온라인 환경에서 해당 오프라인 알고리즘의 동작을 시뮬레이션 한다. OzaBag, OzaBoost에 각각 ADWIN을 사용하여 Drift를 감지한 알고리즘을 Adwin Online Bagging (Adwin Oza Bag)[25], Adwin Online Boost(Adwin Oza Boost)라고 한다.

4) LevBag(Leveraging bagging)[26] : Oza Bag 기반으로 ADWIN을 추가한 하드 코딩된 Drift 감지기 이다. Poisson 분포의 다양성 값(λ=6)을 증가시켜 instance에서 훈련할 확률을 줄여 준다.

5) DDD [27] : Online Bagging의 변형으로 Drift 감지 전후 다양성이 각기 다른 네 가지 ensemble을 상용한다. 사용된 ensemble은 속도와 concept의 길이에 따라 점진적 Drift가 어떻게 나타나는지 분석 한다.

6) FASE(Fast adaptive stack- ing of ensembles)[28] : OzaBag 기반의 알고리즘이며, 메타 분류기를 통해 ensemble 기본 구성을 처리하고 Drift를 감지한다.

7) Online Boost-by-majority (Online BBM) [29] : Freund's BBM algorithm [30]의 Online 버전이다. 중요도에 따른 가중 학습보다 중요도가 약한 학습으로도 유사한 결가를 도출하는 방안을 제안한다.

8) ADOB [31] : Oza Boost를 기반으로 하는 Boosting 앙상블로, 분류기 분산시 영향을 미치는 각 instance를 정확도에 따라 정렬 처리 한다.

3. Concept drift에의한 ML 모델 성능 변화의 정량적 추정 방법

3.1 개요

trainset으로 모델을 학습한 이후, 시간이 경과되면, live data는 어떠한 사회, 문화, 환경적 요인에 의해 그 특성과 분포가 변하게 된다.

학습 이후 이러한 현상은 Data drift, Concept drift로 설명할 수 있으나, 이를 학습시점의 상황으로 보면, trainset과 testset의 분포가 상이한 Covariate shift[32] 상태로 설명할 수 있다. 따라서, Covariate Shift 상태의 학습 방법에 대한 연구들을 통해, 모델 성능 추정 방법을 도출하고자 한다.

Covariate shift 문제에서 trainset은 Equation (1)로 정의 한다.

$$X_L = \mathrm{x}_{1,}\cdots,\mathrm{x}_{\mathrm{m}}, \mathrm{Y}_{\mathrm{L}} = \mathrm{y}_{1,}\cdots,\mathrm{y}_{\mathrm{m}} \tag{1}$$

Equation (1)에서 trainset x의 distribution은 $p(x|\lambda)$ 이다. 그리고, testset은 Equation (2)와 같이 정의 한다.

$$X_T = \mathrm{x}_{\mathrm{m}+1},\cdots,\mathrm{x}_{\mathrm{m}+\mathrm{n}}, \mathrm{Y}_{\mathrm{T}} = \mathrm{y}_{\mathrm{m}+1},\cdots,\mathrm{y}_{\mathrm{m}+\mathrm{n}} \tag{2}$$

Equation (2)에서 testset x의 distribution은 $p(x|\theta)$ 이다. covariate shift 상황에서의 학습 방법에 대한 연구[33]에 의하면 Equation (1),(2) 조건에 대한 loss function의 관계식은 Equation (3)과 같이 정의된다.

$$E_{(x,y)\sim\theta}[l(f(x),y)] = E_{(x,y)\sim\lambda}[\frac{p(x|\theta)}{p(x|\lambda)} l(f(x),y)] \tag{3}$$

Equation (3)에서 $l(f(x),y)$은 모델 $f(x)$의 loss function을 의미한다. loss는 scale 조정에 의해 covariate shift을 보정할 수 있는 관계식이 되고, Drift 상황으로 일반화해서 성능 관계식으로 바꿀 수 있다. 이를 통해 $f(x)$를 일반적인 수치 예측(regression)이라 가정하면 Equation (4)와 같이 표현된다.

$$l_{squared}(f(x),y) = (y-f(x))^2 \tag{4}$$

Equation (3)에 Equation (4)를 대입해서 정리하면, Equation (5)로 표현되고,

$$E_{(x,y)\sim\theta}[(y-f(x))^2] = E_{(x,y)\sim\lambda}[\frac{p(x \mid \theta)}{p(x \mid \lambda)}(y-f(x))^2] \tag{5}$$

각 dataset의 distribution 비율과 모델의 loss는 상호 독립이라 할 수 있으며 Expectation 연산 정리[34]에 의해 $E(AB) = E(A)E(B)$를 대입하면, Equation (6)으로 나타낼 수 있다.

$$E_{(x,y)\sim\theta}[(y-f(x))^2] = E_{(x,y)\sim\lambda}[\frac{p(x|\theta)}{p(x|\lambda)}]E_{(x,y)\sim\lambda}[(y-f(x))^2] \tag{6}$$

이때 각각의 값은 아래와 같이 표현된다.

$E_{(x,y)\sim\theta}[(y-f(x))^2] = Testset$에 대한 $MSE(Mean\,Square\,Error)$

$E_{(x,y)\sim\lambda}[(y-f(x))^2] = Trainset$에 대한 $MSE(Mean\,Square\,Error)$

이를 Equation (6)에 대입 정리하면 Equation (7)로 표현이 가능하다.

$$MSE_{testset} = E_{(x,y)\sim\lambda}[\frac{p(x|\theta)}{p(x|\lambda)}]MSE_{trainset} \tag{7}$$

Equation (7)을 통해, testset(live data)과 trainset(최초 측정 성능)의 성능 및 distribution의 관계식이 성립함을 알 수 있다.

이러한 이론적 배경을 기반으로 본 연구에서는 concept drift의 통계량을 통해 직접적으로 precision 성능을 예측하는 성능 예측 모델(PPM, Performance prediction model)을 제안한다.

3.2 제안 방법

성능 예측 대상 모델을 $f(x,L)$이라 할 때, trainset과 testset은 $f(x,L)$의 학습 및 성능 측정을 위해 활용된 실 데이터에 해당 된다. 그리고 testset을 통하여 $f(x,L)$의 성능 base score가 측정된다.

각 단계별 제안 방법은 아래와 같이 설명한다.

1) Step1 : 성능 모델링 단계(performance modeling)

Algorithm 1. Step1 : performance modeling

1: $L_{trainset}=\{Y_L,X_L\}$ #이하 간략히 L 로 표기
2: $L_{testset}=\{Y_T,X_T\}$
3: $f(x,L)=train(Y_L,X_L)$
4: $base\_score=score(f(X_T,L),Y_T)$
5: *For number_of_sample do*
6: $Y_b,X_b=block\_sampling(Y_L,X_L,sampling\_ratio)$
7: $X_b=noise\_injection(X_b,noise\_intensity)$
8: $drift=adjusted\_ks\_test(X_L,X_b)$
9: $\triangle score\_array \leftarrow (base\_score-score(f(X_b,L),Y_b))$
10: $drift\_array \leftarrow drift$
11: *end*
12: $ppm(drift)=train(\triangle score\_array, drift\_array)$

a) trainset에 대해 주어진 sampling ratio의 비율 범위내 random size로 블럭 샘플링 (block sampling)을 한다.
b) 샘플링된X_b 일부에 '표준편차(X_L)·noise intensity'을 표준편차로 갖는 Normal distribution에서 임의 취득한 값을 더하여 X_L의 특성을 반영한 drift를 유도한다.
c) trainset(X_L)과 drift유도된 Sample(X_b)에 대해 변경된 Kolmogorov-Smirnov test를 하여, drift 통계량(Kolmogorov-Smirnov statistic)을 취득한다.
d) 이때, 변경된 KS Test는 기존의 KS Test의 Equation (8)에서 Equation (9)와 같이 변경하여 적용한다.

$$D_n=sup_x|F_n(x)-F(x)| \quad (8)$$

$$D_n=sup_x(F_n(x)-F(x)) \quad (9)$$

e) Equation (9)와 같이 절대값를 제거하므로 drift 통계량의 선형 특성을 살려서 성능 예측에 대한 drift 통계량의 설명력을 높인다.
f) drift유도된 Sample에 대하여 $f(x,L)$의 성능 감소차($\triangle score$)를 $\triangle score_array$에 저장한다.
g) drift통계량을 $drift_array$에 저장한다.
h) Equation (1)~(7)까지의 과정을 반복하여 drift에 대한 성능 감소차 dataset를 생성한다.
i) drift 성능 감소차 dataset을 통해, $ppm(drift)$을 학습한다.

2) Step2 : 성능 감소차 보정 단계(adjust value)

Step1에서 X_b 일부를 임의 치환하여 drift를 유도하였으므로 drift에 의한 성능 감소차($\triangle score$)는 실제 drift에서 발생하는 감소차이와 다를 수 있다. 이를 조정하기 위해 testset을 통해, 그 감소차 비율을 측정한다.

Algorithm 2. Step2 : adjust value

1: *For number_of_sample do*
2: $Y_b,X_b=block\_sampling(Y_T,X_T,sampling\_ratio)$
3: $drift=adjusted\_ks\_test(X_L,X_b)$
4: $\triangle score\_array_{predicted} \leftarrow ppm(drift)$
5: $\triangle score\_array_{real} \leftarrow (base\_score-score(f(X_b,L),Y_b))$
6: *end*
7: $adjusted\_value=\frac{range(\triangle\ score\_array_{real})}{range(\triangle\ score\_array_{predicted})}$

a) testset(Y_T,X_T)에 대하여 주어진 sampling ratio의 비율 범위내에서 random size 블럭 샘플링(block sampling)을 한다.
b) trainset과 testset에서 추출된 Sample에서 변경된 Kolmogorov-Smirnov test를 하여, drift통계량을 취득한다.
c) Step1에서 학습된 $ppm(drift)$을 통해, 예측 성능 감소차 $\triangle score_{predicted}$를 구해 $\triangle score_array_{predicted}$에 저장한다.
d) testset에서 추출된 Sample(Y_b,X_b)에 대해 $f(x,L)$의 실제 성능 감소차($\triangle score_{real}$)를 구하여 $\triangle score_array_{real}$에 저장한다.
e) Equation (1)~(4)의 과정을 반복하여 예측 성능 감소차 dataset과 실 성능 감소차 dataset을 생성한다.
f) 예측 성능 감소차의 범위(Max-Min)와 실제 성능 감소차의 범위에 대한 비율을 보정 비율값으로 구한다.

3) Step3 : $ppm(drift)$적용 단계(apply performance model)

Algorithm 3. Step3 : apply performance_model

1: $l_{Input\_live\_data}=\{X_l\}$#live data에는 Y가 미존재
2: $drift=adjusted\_ks\_test(X_L,X_l)$
3: $score_{predicted}$
$=base\_score-adjusted\_value*ppm(drift)$

a) 일정기간 축적된 운영실제 dataset (X_l)를 취득한다.
b) trainset(X_L)과 실제 dataset (X_l)에 대해 변경된 Kolmogorov-Smirnov test를 하여, drift 통계량을

취득한다.

c) $ppm(drift)$을 통해 예측 성능 감소차($\triangle score_{predicted}$)를 구하고 보정값을 곱한 후 base score의 차이값으로 예측 성능값($score_{predicted}$)을 구한다.

4. 실 험

4.1 개요

본 논문에서 제안하는 성능 예측 모델(PPM, Performance prediction model)의 범용적(robust) 적용 가능성을 검증하기 위해 2개의 모델을 기반으로 실험을 진행하여 검증한다.

4.2 실험 계획

1) 대출 심사 모델의 성능 추정 실험

a) Personal Loan Data

- 미국 개인대출 기업 'Lending Club'의 공개 데이터
- 2007~2017년까지 총 2,260,701건의 데이터
- 대출 건당 실제 'default' 여부 포함 , 151개 Column,
- 데이터 정제 후, 81개 Features, 120만건 사용

trainset : 11,497건 (2010년)
online dataset : 1,268,183건 (2011~2017년)

b) 대출 심사 알고리즘 : Random-forest

c) 성능 예측 알고리즘 : Ordinary Least Squares regression

d) 실험의 변수

- feature_count (order of importance)
- sampling ratio
- noise intensity

e) 성능 지표 : precision

2) 신용카드 오사용 탐지 모델의 성능 추정 실험

a) Credit Card Transaction Data

- IBM multi-agent virtual world simulation 데이터
- 약 2천만 건 이상의 transaction 포함
- 미국에 거주하는 약 2,000명의 소비자
- 수십 년의 구매 및 여러 카드 정보 포함

- trainset : 37,301건 (2007년, 1/30 축소)
- online dataset : 90,836건 (2008~2009년, 1/30 축소)

b) 신용카드 오사용 탐지 알고리즘 : Catboost

c) 성능 예측 알고리즘 : Ordinary Least Squares regression

d) 실험의 변수

- feature_count (order of importance)
- sampling ratio
- noise intensity

e) 성능 지표 : precision

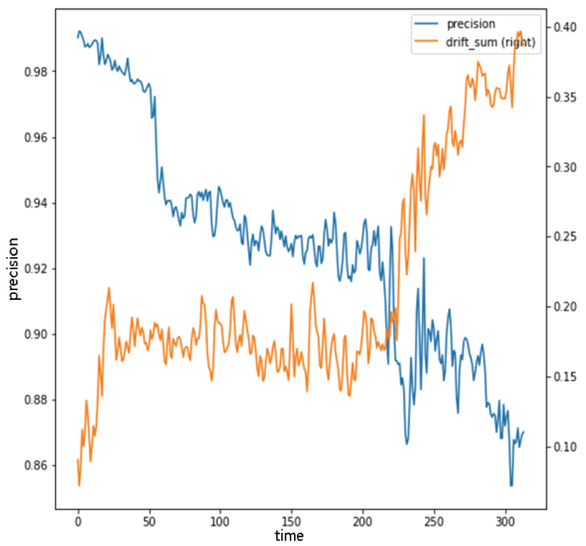

Fig. 2. The Precision of the Model and the Sum of Drift Statistics (for 3 features)

4.3 실험 결과

1) 대출 심사 모델 실험 결과

a) concept drift 측정 결과

importance가 높은 상위 3개의 feature('recoveries', 'last_fico_range_high', 'last_pymnt_amnt')를 선택하여 KS-Test 통계량의 합과 precision을 비교하였다.

Fig. 2에서 각 feature의 drift 통계량 합이 증가하는 것을 확인할 수 있다. 그리고 drift 통계량 합이 증가할 때 반대로 precision은 낮아지는 것을 확인할 수 있다.

b) 성능 추정 실험 결과

Table 1. Loan Predict Experimental Results

실험	feature count	sampling ratio	noise intensity	R^2
#1.1	2	0.05~0.95	1.0	-0.6296 ±0.0564
#1.2	3			0.2803 ±0.0771
#1.3	4			0.3135 ±0.0583
#1.4	5			-0.1628 ±0.3067
#1.5	6			0.5237 ±0.0736
#1.6	7			0.5345 ±0.0886
#1.7	8			0.1314 ±0.3974
#1.8	9			0.0556 ±0.3681
#1.9	10			-0.0785 ±0.5655
#1.10	**7**	**0.10~0.90**	**1.0**	**0.5574 ±0.0848**
#1.11		0.15~0.85		0.5497 ±0.0643
#1.12		0.20~0.80		0.5021 ±0.1097
#1.13		0.25~0.75		0.4910 ±0.1058
#1.14		0.30~0.70		0.4615 ±0.1236
#1.15	7	0.10~0.90	1.5	0.5517 ±0.0818
#1.16			2.0	0.5313 ±0.0608
#1.17			2.5	0.5123 ±0.0651
#1.18			3.0	0.4956 ±0.0758

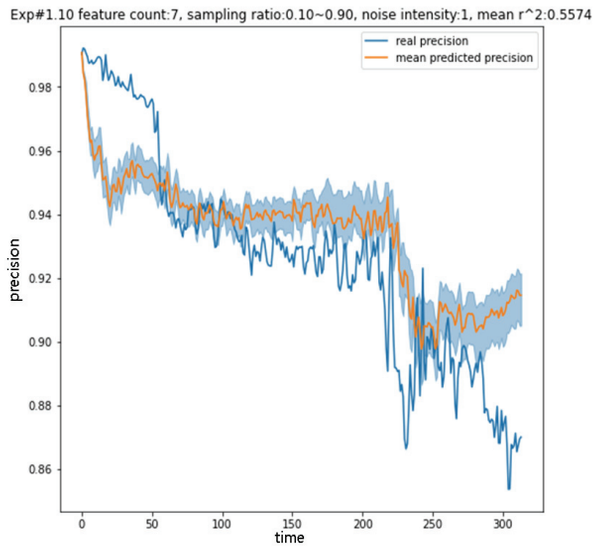

Fig. 3. Comparison of Actual Precision and Estimated Precision

실험1.10, feature_count 7, sampling_ratio 0.10~ 0.90, noise intensity 1.0의 경우, R^2 0.5574 ±0.0848 으로 가장 높게 나타났다. 총 81개의 feature 중 약 8.6%의 feature가 적용되었다.

실험1.10에서의 실제 precision과 추정 precision의 비교 차트는 Fig. 3과 같다.

Fig. 3에서 추정 precision의 위, 아래 색이 채워진 구간은 추정의 최대값과 최소값 구간이며 추정의 오차범위라 할 수 있다. 추정에 사용한 feature count의 개수는 추정의 정확도에 많은 영향을 주었으며, sampling ratio와 noise intensity는 feature count에 비해 상대적으로 낮은 영향을 주었다. R^2 0.5대의 낮은 결과이지만, 주요한 패턴의 추정이 가능한 것을 확인할 수 있다.

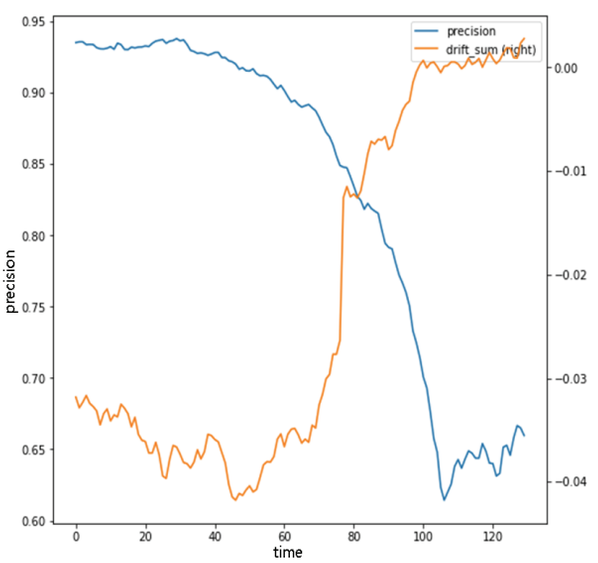

Fig. 4. The Precision of the Model and the Sum of Drift Statistics (for 2 features)

2) 신용카드 오사용 탐지 모델 실험 결과

a) concept drift 측정 결과

importance가 높은 상위 2개의 feature('MMC', 'Hour')를 선택하여 KS-Test 통계량의 합과 precision을 비교하였다.

Fig. 4에서 각 feature의 drift 통계량 합이 증가하는 것을 확인할 수 있다. 그리고 drift 통계량 합이 증가할 때 반대로 precision은 낮아지는 것을 확인할 수 있다.

b) 성능 추정 실험 결과

Table 2. Credit Card Experiment Results

실험	feature count	sampling ratio	noise intensity	R2
#2.1	1	0.05~0.95	1.0	0.6564 ±0.1290
#2.2	2			0.6905 ±0.0201
#2.3	3			0.5929 ±0.0411
#2.4	4			0.6279 ±0.0202
#2.5	5			0.6339 ±0.0413
#2.6	2	0.10~0.90	1.0	0.6931 ±0.0191
#2.7		0.15~0.85		0.6985 ±0.0148
#2.8		0.20~0.80		0.7029 ±0.0155
#2.9		0.25~0.75		0.7149 ±0.0200
#2.10		0.30~0.70		0.7169 ±0.0181
#2.11		0.35~0.65		0.7176 ±0.0175
#2.12	**2**	**0.35~0.65**	1.5	0.7215 ±0.0174
#2.13			2.0	0.7270 ±0.0148
#2.14			**2.5**	**0.7278 ±0.0143**
#2.15			3.0	0.6531 ±0.1317

실험2.14, feature count 2, sampling ratio 0.35~ 0.65, noise intensity 2.5의 경우, R^2 0.7278 으로 가장 높게 나타났다. 총 7개의 feature 중 약 28.5%의 feature가 적용되었다.

실험2.14에서 실제 precision과 추정 precision의 비교 차트는 Fig. 5와 같다.

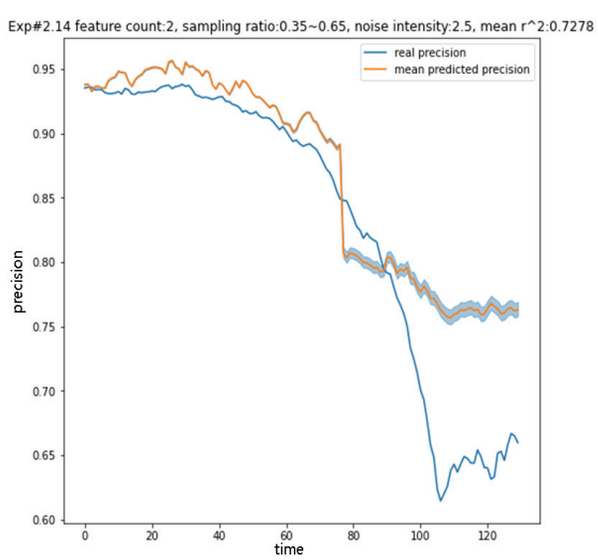

Fig. 5. Comparison of Actual Precision and Estimated Precision

추정에 사용한 feature count의 개수는 추정의 정확도에 많은 영향을 주었으며, sampling ratio와 noise intensity는 feature count에 비해 상대적으로 낮은 영향을 주었다. R^2 0.7대의 낮은 결과이지만, 주요한 패턴의 추정이 가능한 것을 확인할 수 있다.

5. 결 론

두 개의 비즈니스 모델을 통한 실험을 통해, Ground Truth가 없는 Live data (online data)에 대한 모델의 성능 추정이 가능함을 확인하였다. R2는 0.5~0.7대의 낮은 수치이나 기존의 연구에서는 시도되지 않았던 기계학습 모델의 운영 중 성능 추정이 가능함을 확인한 점에서 그 의미가 크다. 제안한 알고리즘에서 sample ratio나 noise intensity는 예측 성능에 크게 영향을 미치지는 않았다. 가장 중요한 것은 '어떤 feature의 drift를 대상으로 할 것이냐'이다. 본 연구에서는 대상 feature를 선정하는 방법론까지 제안하지 않았으나 향후 연구를 통해 feature 선택 방법 및 변수 최적화 방법, 현실 데이터의 변화 양상과 feature의 특성을 반영한 drift 유도 방법 등을 찾는다면 더욱 정교한 성능 예측이 가능할 것이다. 그리고 본 연구에서 제안한 성능 예측 모델이 drift의 유형(sudden drift, gradual drift, incremental drift 등)에 따라 어떻게 적용될 수 있을지 추가 연구가 필요하다.

References

[1] BHOWMIK, Pritom. Machine Learning in Production: From Experimented ML Model to System. 2022.

[2] Machine Learning Monitoring [Internet], https://evidentlyai.com/blog/machine-learning-monitoring-data-and-oncept-drift

[3] Data Drift vs. Concept Drift [Internet], https://deepchecks.com/data-drift-vs-concept-drift- what-are-the-main-differences/

[4] M. Salganicoff, "Tolerating concept and sampling shift in lazy learning using prediction error context switching," *Artificial Intelligence Review*, Vol.11, No.1-5, pp.133-155, 1997.

[5] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," *Advances in Neural Information Processing Systems*, Vol.32, 2019.

[6] L. I. Kuncheva, "Classifier ensembles for changing environments," In *Multiple Classifier Systems: 5th International Workshop, MCS 2004, Cagliari, Italy, June 9-11, 2004. Proceedings 5*. Springer Berlin Heidelberg, pp.1-15, 2004.

[7] L. Kuncheva, "Classifier ensembles for detecting concept change in streaming data: Overview and perspectives," In *Proceedings of the 2nd Workshop SUEMA 2008*. 2008.

[8] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Advances in Artificial Intelligence: SBIA 2004, in: vol. 3171 of (LNCS)*, Springer, pp.286-295, 2004.

[9] M. Baena-Garcia, J. Del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, and R. Morales-Bueno, "Early drift detection method," in *International Workshop on Knowledge Discovery from Data Streams*, pp.77-86, 2006.

[10] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 7th SIAM International Conference on Data Mining (SDM'07)*, Minneapolis, MN, USA, pp.443-448, 2007.

[11] R. Pears, S. Sakthithasan, and Y. Koh, "Detecting concept change in dynamic data streams," *Machine Learning*, Vol.97, No.3, pp.259-293, 2014.

[12] S. Bernstein, "The theory of probabilities," Gastehizdat Publishing House, Moscow, 1946.

[13] D. T. J. Huang, Y. S. Koh, G. Dobbie, and R. Pears, "Detecting volatility shift in data streams," in *Proceedings of 2014 IEEE International Conference on Data Mining ICDM*, Shenzhen, China, pp.863-868, 2014.

[14] K. Nishida and K. Yamauchi, "Detecting concept drift using statistical testing," in *Proceedings of 10th International Conference on Discovery Science (DS'07)*, in: Vol.4755 of (LNCS), Springer, pp.264-269, 2007.

[15] R. S. M. Barros, "Advances in data stream mining with concept drift," Professorship (Full) Thesis, Centro de Informática, Universidade Federal de Pernambuco, Brazil, 2017.

[16] R. S. M. Barros, J. I. G. Hidalgo, and D. R. L. Cabral, "Wilcoxon rank sum test drift detector," *Neurocomputing*, Vol.275, pp.1954-1963, 2018.

[17] D. R. L. Cabral, "Statistical tests and detection of concept drifts in data streams," Centro de Informática, Universidade Federal de Pernambuco., Portuguese., 2017 (M.Sc. Dissertation).

[18] D. R. L. Cabral and R. S. M. Barros, "Concept drift detection based on fisher's exact test," *Information Sciences,* Vol.442, pp.220-234, 2018.

[19] R. Fisher, "On the interpretation of χ^2 from contingency tables, and the calculation of P," *Journal of the Royal Statistical Society,* Vol.85, No.1, pp.87-94, 1922.

[20] J. Z. Kolter and M. A. Maloof, "Dynamic weighted majority: an ensemble method for drifting concepts," *The Journal of Machine Learning Research,* Vol.8, pp.2755-2790, 2007.

[21] A. Blum, "Empirical support for winnow and weighted-majority algorithms: Results on a calendar scheduling domain," *Machine Learning,* Vol.26, No.1, pp.5-23, 1997.
[22] L. Breiman, "Bagging predictors," *Machine Learning,* Vol.24, No.2, pp.123-140, 1996.
[23] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning,* Vol.96, pp.148-156, 1996.
[24] N. C. Oza and S. Russell, "Online bagging and boosting," in *Artificial Intelligence and Statistics, Morgan Kaufman*, pp.105-112, 2001.
[25] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams, in *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (KDD'09)*, Paris, France, pp.139-148, 2009.
[26] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Machine Learning and Knowledge Discovery in Databases, in: Vol.6321 of (LNCS)*, Springer, pp.135-150, 2010.
[27] L. L. Minku and X. Yao, "DDD: a new ensemble approach for dealing with concept drift," *IEEE transactions on Knowledge and Data Engineering,* Vol.24, No.4, pp.619-633, 2012.
[28] I. Frías-Blanco, A. Verdecia-Cabrera, A. Ortiz-Díaz, and A. Carvalho, "Fast adaptive stack- ing of ensembles," in *Proceedings of the 31st ACM Symposium on Applied Computing (SAC'16)*, Pisa, Italy, pp.929-934, 2016.
[29] A. Beygelzimer, S. Kale, and H. Luo, "Optimal and adaptive algorithms for online boosting," *International Conference on Machine Learning. PMLR*, 2015.
[30] Y. Freund, "Boosting a weak learning algorithm by majority," *Information and Computation,* Vol.121, No.2, pp.256-285, 1995.
[31] S. G. T. C. Santos, P. M. Gonçalves Jr., G. D. S. Silva, and R. S. M. Barros, "Speeding up recovery from concept drifts," in *Machine Learning and Knowledge Discovery in Databases, in: Vol. 8726 of (LNCS)*, Springer, pp.179-194, 2014.
[32] N. G. Nair, P. Satpathy, and J. Christopher, "Covariate shift: A review and analysis on classifiers," In *2019 Global Conference for Advancement in Technology (GCAT)*. IEEE. pp.1-6, 2019.
[33] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, Vol.90, No.2, pp.227-244, 2000.
[34] Expectation and Variance [Internet], https://www.stat.auckland.ac.nz/~fewster/325/notes/ch3blank.pdf

안 순 홍

https://orcid.org/0000-0003-4500-9183
e-mail : kingmir@dankook.ac.kr
2004년 단국대학교 전자계산(학사)
2006년 단국대학교 컴퓨터과학(석사)
2014년 단국대학교 컴퓨터공학(박사수료)
2006년 ~ 현 재 아시아나IDT AI빅데이터연구소 수석연구원
관심분야 : 분산알고리즘, 유무선망에서의 라우팅, 유무선망에서의 멀티미디어 통신, 기계학습

이 훈 석

https://orcid.org/0000-0003-4775-2720
e-mail : dolmani38@naver.com
2006년 건국대학교 물리학(학사)
2009년 연세대학교 전파통신공학과(석사)
2014년 단국대학교 컴퓨터공학(박사수료)
2009년 ~ 2006년 LG히다찌 주임연구원
2006년 ~ 현 재 아시아나IDT AI빅데이터연구소 수석연구원
관심분야 : 기계학습, 자연어처리, RFID 미들웨어

김 승 훈

https://orcid.org/0000-0001-5021-3627
e-mail : edina@dankook.ac.kr
1985년 인하대학교 전자계산학(학사)
1989년 인하대학교 전자계산학(석사)
1998년 포항공과대학교 컴퓨터공학(박사)
1989년 ~ 1990년 한국전자통신연구원 연구원
1990년 ~ 1993년 포스데이타(주) 정보통신본부 연구원
1998년 ~ 2001년 상지대학교 조교수
2001년 ~ 현 재 단국대학교 컴퓨터공학과 교수
관심분야 : 분산알고리즘, 유무선망에서의 라우팅, 유무선망에서의 멀티미디어 통신, 기계학습