**Performance Evaluation for**

**SPT Algorithm**

|  |
| --- |
|  |

**응용통계학과**

**주건재**

|  |
| --- |
|  |

<  목    차  >

1 서론 1

1.1분석개요 1

1.2 분석목표 2

1.3 데이터소개 2

2 본론 3

2.1 분석 시나리오 3

2.2 SPT 4

3 결론 6

3.1 분석결과 및 토의 6

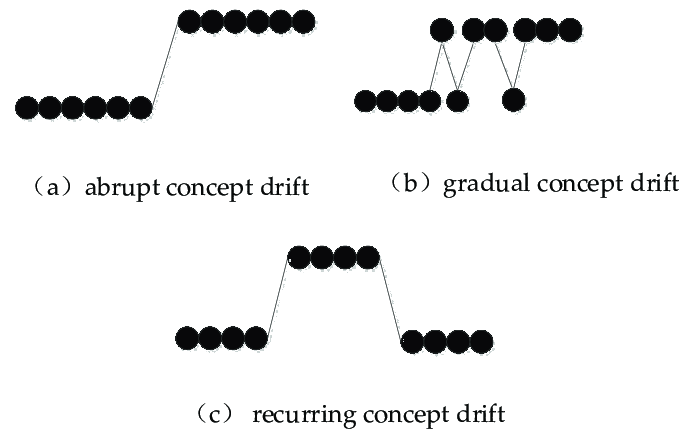
3.2 결론 및 추후 연구 6

4 REFERENCES 7

1. **서론**
   1. **분석개요**

많은 산업에서 스트리밍 데이터를 사용하여 머신러닝 파이프라인을 통해 실시간 데이터 분석 및 예측을 한다. 스트리밍 데이터를 사용하는 환경에서는 데이터 전체의 특성을 파악하고 모델링을 진행할 수 없다. 그렇기에 데이터 특성의 변화를 감지하고 그에 맞춰 변화하는 동적인 모델을 사용하여야 한다.

Concept Drift는 데이터 변화 중 하나이다. 이는 의 변화를 얘기한다. 즉 x값는 상관없이 종속변수 y 값이 변하는 것이다. 원인은 데이터를 수집하는 환경의 변화 일 수도 있고 데이터 자체의 변화 일 수도 있다. <그림1>은 Concept Drift의 세가지 종류를 나타낸다. (a) abrupt concept drift는 데이터의 급격한 변화, (b) gradual concept drift는 점진적인 변화, (c) recurring concept drift는 일시적인 변화를 뜻한다. 본 연구에서는 abrupt concept drift가 발생하는 데이터를 사용하여 Drift를 대처하는 알고리즘 SPT(Self hyper-Parameter-Tuning)을 적용할 것이다.



<그림1> Concept Drift의 종류[[1]](#footnote-1)

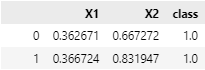
* 1. **분석 목표**

Abrupt Concept Drift가 존재하는 생성된 데이터에 SPT 알고리즘을 적용하고 적용여부에 따른 성능차이를 제시한다.

첫째 목표: SPT 알고리즘을 구현하고 적용한다.  
둘째 목표: 적용 여부에 따른 성능차이를 보여준다.

* 1. **데이터소개**

본 데이터는dataverse에서 오픈소스로 제공한다[[2]](#footnote-2).



<표1> Abrupt Concept Drift 데이터 샘플

<표1>는 Abrupt Concept Drift를 인위적으로 생성된 데이터이다. 독립변수는 X1, X2 두개로 -1에서 1까지의 값을 갖는다. 종속변수는 “class” 변수로 0과 1을 갖는다. X1과 X2는 독립적이고 종속변수는 두 클래스의 수가 동일하다. 데이터의 샘플 수는 40000개이고 10000, 20000, 30000번째 Abrupt Concept Drift가 일어난다.

1. **본론**

**2-1. 분석 시나리오**

본 연구에서는 몇 가지 시나리오를 만들고 비교 평가하여 SPT 알고리즘이 필요한지 알아볼 것이다.

1. Batch Learning without Drift Scenario

시나리오 1에서는 데이터의 마지막 10000개 샘플을 사용한다. Concept Drift가 없는 상황으로 7000개의 훈련 데이터와 마지막 3000개의 테스트 데이터로 나눠서 진행한다. Hyper-parameter를 튜닝하기 전과 후로 나눠서 진행한다. 이는 Hyper-parameter를 튜닝의 효과 여부를 확인한다.

1. Batch Learning with Drift Scenario

시나리오 2 에서는 전체 40000개의 데이터를 사용한다. 앞의 37000개를 훈련 데이터로 마지막 3000개를 테스트 데이터로 사용한다. 시나리오2는 시나리오1의 결과와 비교하여 Concept Drift가 모델 성능에 어떤 영향을 주는지 대해 알아보기 위함이다.

1. Streaming Learning without self-hyper-parameter tuning

시나리오3 에서는 40000개 전체 데이터를 사용하며 훈련/테스트 데이터로 나누지 않는다. 본 시나리오는 Concept Drift가 존재하는 데이터에 아무런 조치를 하지 않고 모델을 사용하면 어떤 결과가 나오는지 확인한다.

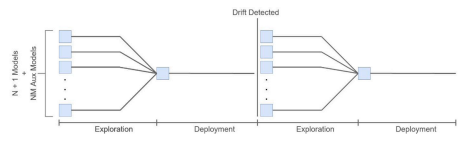
1. Streaming Learning with self-hyper-parameter tuning

시나리오4 에서는 시나리오3과 같은 환경과 base model(베이스 머신러닝 모델)로 SPT 알고리즘을 적용하여 모델링을 했을 때 시나리오3의 결과와 어떻게 상이한지 확인한다.

평가지표로는 이진분류에 통상적으로 사용되는 f1-score와 accuracy를 사용하여 각 시나리오의 모델 별 결과를 비교할 것이다.

* 1. **SPT**

1. **SPT**



<그림2> SPT Algorithm 도식화[[3]](#footnote-3)

<그림2>는 SPT 알고리즘의 전체 흐름을 도식화한 것이다. SPT는 Exploration단계와 Deployment 단계와 나누어진다. Exploration 단계에서는 hyper-parameter를 tuning하는 단계로 기본적인 최적화 알고리즘 중 하나인 Nelder-Mead를 실시간 모델에 맞는 형태로 바꾸어서 사용한다. 모델의 최적화를 판단하는 Convergence Rule에 의하여 최적화 완료 여부를 판단하고 완료가 되면 Deployment단계로 변경한다.

Deployment 단계는 Exploration단계에서 hyper-parameter가 최적화된 모델을 사용한다. Streaming 데이터로 지속적으로 예측 및 에러를 계산하여 Concept Drift가 발생하는지 모니터링한다. 에러 값을 토대로 Concept Drift를 모니터링하는 Drift Detection Method 알고리즘[[4]](#footnote-4)을 사용한다. Drift가 발생하면 다시 Exploration 단계로 변경한다.

1. **Exploration**

<그림3>은 Exploration단계의 세부적인 도식이다. 파란 상자는 N+1개의 후보 모델이다. 후보 모델들은 N개의 hyper-parameter를 가지며 다 다른 값을 가지고 있다. 수직선은 간의 거리는 데이터 샘플의 양을 나타내며 이는 추후 설명할 Dynamic Sample Size이다. 수직선은 모델들의 한번 업데이트를 의미하며 이는 Online Nelder-Mead 알고리즘에 의해 결정되며 각모델들의 RMSE값이 관여한다. 마지막 수직선은 Convergence 수직선으로 모델의 hyper-parameter 최적화가 완료된 상태이다.

**2-1) Online Nelder-Mead**

Nelder-Mead 최적화 알고리즘은 어떠한 함수의 최소값을 미분을 사용하지 않고 찾는 알고리즘이다.[[5]](#footnote-5) 파라미터 최적화의 경우 그 함수가 목적함수가 되고 SPT의 경우 RMSE값이 된다.

Online Nelder-Mead는 <그림3>의 수직선 간의 데이터 샘플을 사용하여 RMSE의 함수f를 산출한다. 이 함수로 Nelder-Mead 방법에 적용하여 모델을 한번 업데이트한다. 이후 이를 Convergence Rule에 부합할 때까지 반복한다. Online 버전의 특징은 함수f가 수직선 구간마다 샘플 데이터에 의해 변하는 것이다.

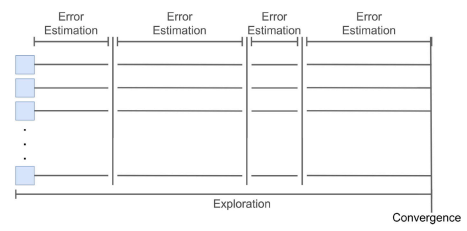
**2-2) Dynamic Sample Size**

<식2> SPT Dynamic Sample Size

<식2>는 SPT에서 제안하는 Exploration단계의 동적 샘플사이즈 식이다. σ는 N+1개의 모델에서 산출한 RMSE의 표준편차이다. M은 Desired Error Margin으로 0.95를 논문에서 제안한다. 샘플 사이즈가 너무 작아지는 것을 방지하기 위해 최소값은 30으로 한다.

**2-3) Convergence Rule**

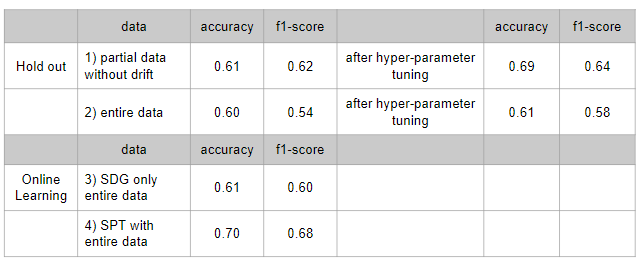
Online Nelder-Mead의 가장 성능이 좋은 모델, 그 다음 좋은 모델 그리고 가장 안 좋은 모델의 가운데 다음 업데이트 될 모델이 위치하면 모델이 Converged 한 것이다. 모델이 Converged 하면 Deployment 단계로 넘어간다.



<그림3> SPT Exploration단계 세부 도식

1. **결론**

**3-1. 분석결과 및 토의**



<표3> 모델 분석결과 비교표

<표13>은 각 모델들의 결과이다. 베이스 모델은 Logistic Regression을 사용하였다. 튜닝하는 hyper-parameter는 L2규제를 사용하였다. Hold Out의 결과를 보면 hyper-parameter는 모델을 훈련하는데 유의한 성능 개선을 주는 것을 볼 수 있었고 Concept Drift가 있는 시나리오2번의 결과가 Drift가 없는 시나리오1 번의 결과보다 안 좋은 것을 확인했다.

Streaming Learning인 Online Learning의 경우 Concept Drift에 대비한 시나리오4의 모델이 그렇지 않은 시나리오3의 모델보다 좋은 것을 확인하였다. 또한, 시나리오1의 모델과 시나리오4의 결과를 볼 때 Streaming Learning이 Batch Learning에 비해 결과가 나쁘지 않다는 것도 볼 수 있었다. 이러한 결과로 미루어 보아 Concept Drift가 있을지 없을지 모르는 Streaming Data의 분석은 Concept Drift를 대비해야 하고 SPT는 그 중 하나의 방법이 될 수 있다.

**3-2. 연구의 결론 및 한계**

본 연구에서 SPT알고리즘을 구현하고 이를 Concept Drift Data에 적용해 보았다. 결과적으로 베이스 모델에 비해 성능에 향상을 보였다. Abrupt Concept Drift 이외에 다른 두가지 Concept Drift에 대해 연구를 해보고 더욱 나아가 다른 Drift에도 적용해보면 의미가 있을 것이다.

연구의 가장 큰 한계는 데이터에 있다. 데이터가 인위적으로 생성된 데이터이기 때문에 실제 수집된 데이터에는 어떠한 결과가 나올 수 있을지 보장하기가 힘들다. 그러므로 추후에 실제 데이터를 적용해 보면 유의미한 결과를 얻을 수 있다.

**REFRENCE**

[1] Veloso, B., Gama, J., Malheiro, B., & Vinagre, J. (2021). Hyperparameter self-tuning for data streams. *Information Fusion*, *76*, 75-86.

[2] D. Maclaurin, D. Duvenaud, R.P. Adams, Gradient-based hyperparameter optimization through reversible learning, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning, Vol. 37, ICML’15, JMLR.org, 2015, pp. 2113–2122

[3] J. Gama, P. Medas, G. Castillo, P.P. Rodrigues, Learning with drift detection, in: A.L.C. Bazzan, S. Labidi (Eds.), Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian, in: Lecture Notes in Computer Science, vol. 3171, Springer, 2004, pp. 286–295,

1. https://www.researchgate.net/figure/Types-of-concept-drift\_fig1\_332731962 [↑](#footnote-ref-1)
2. https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/5OWRGB [↑](#footnote-ref-2)
3. [1] Veloso, B., Hyperparameter self-tuning for data streams. *Information Fusion*, [↑](#footnote-ref-3)
4. [3] J. Gama, P. Medas Advances in Artificial Intelligence, [↑](#footnote-ref-4)
5. [2] D. Maclaurin,Gradient-based hyperparameter optimization through reversible learning [↑](#footnote-ref-5)