극소량의 데이터를 활용한 회귀모형 추정 방법

제 19회 한국 대학생 산업공학 프로젝트 경진대회

팀 오작교

김찬호, 변유정, 부도현, 최승준

대한산업공학회

목차

1-1. 탐구동기

“테슬라 사고, 태양 역광 탓” … 자율주행차 또 날씨 오작동

2016년, 테슬라의 자율주행차가 중앙분리대를 들이받아 운전자가 사망하는 사고가 발생함.

이에 테슬라는 사고에 대해
“자동주행 차량이 역광 탓에 흰색 트레일러를 하늘로 오인해 충돌 사고를 냈다 " 고 사고 원인을 밝힘.

악천후로 인한 **예기치 못한 상황**에서 자율주행 차량의 **대처 능력이 현저히 떨어질 수 있다**는 것이 확인됨

센서 감지 데이터의 변화로 인한 잘못된 예측

(a) 캘리포니아의 온화한 시험주행 환경

(b) 사고 당시 강한 역광 하에서의 환경

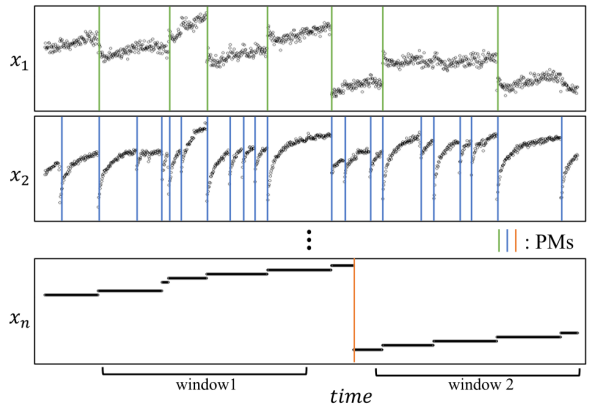
(a) 그동안 자율주행차량을 테스트하는 업체들은 기후가 온화하고 교통체증이나 도로가 복잡하지 않은 미 서부 캘리포니아나 애리조나 지역을 선호해 왔다.

(b) 실제 주행에서는 역광과 같은 요인으로 인하여 기존에 학습하지 못한 익숙치 않은 경우들에 대한 대처가 요구된다.

이처럼 다양한 요인들에 의해 유발되는 **센서 감지 데이터의 변화** 가운데, **정확한 예측**을 수행할 수 있어야 한다.

1-1. 탐구동기

제조현장에서 빈번한 데이터 분포의 변화

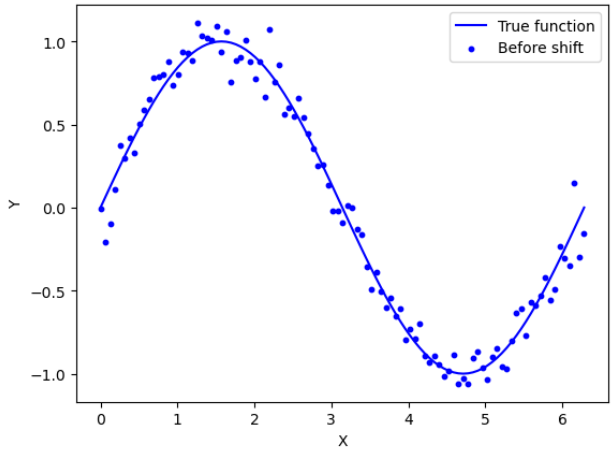


위 그림은, 디스플레이 제조 산업에서 활용되는 ELA(Excimer laser annealing) Process에서 수집된 시간별 센서 데이터를 나타냄.

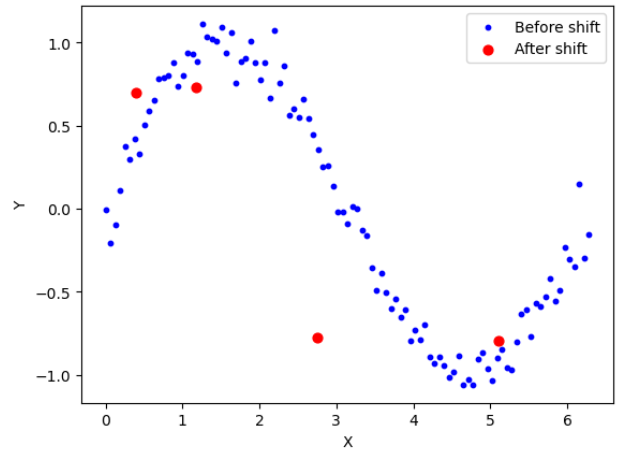
- $\{x_1, x_2, \ldots, x_n\}$: 센서가 수집하는 개별 요소
- 센서별 유지보수(PM)은 데이터 분포의 변화 야기

데이터 분포의 변화 → 기존 예측 모형 적용 X

∴ 새로운 모형 학습비용↑



(a) 분포 변화 이전의 데이터와 True Function

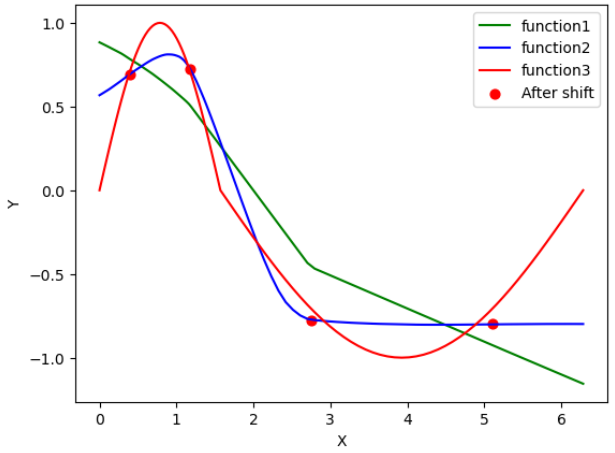


(b) 분포 변화 이전과 이후의 데이터

그림(a)와 같이 기존에는 센서 데이터와 품질이 사인함수 관계를 가지고 있다.

센서의 교체로 인하여 **센서 데이터의 분포 변화**가 발생하였고, (b)와 같이 변화된 분포에서 데이터 4개를 관측하였다.

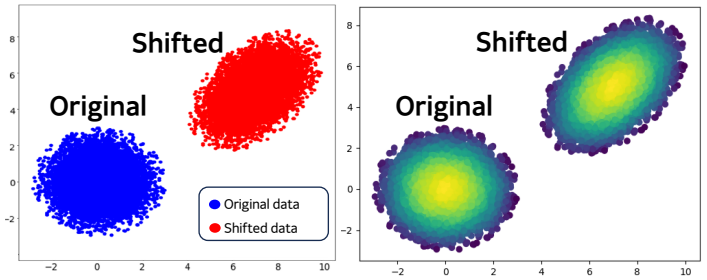
사전지식을 활용하여, 분포 변화 이후 최적의 회귀모형을 **효율적**으로 추정하고자 한다.



(c) 분포 변화 이후의 추정 회귀 모형들

1-2. 제반 선행 지식

Covariate Shift



(a) 입력 분포 변화 전후 (b) 입력 분포 변화 전후의 출력 분포

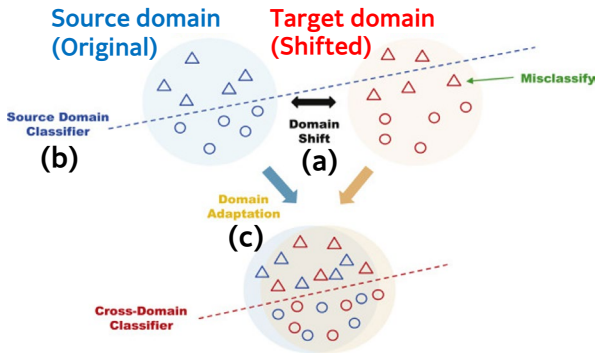
2차원 입력 데이터($\boldsymbol{x}$)에 대해 출력값 $z = 9 - ({x_1}^2 + {x_2}^2)$의 관계를 갖는다고 하자.

제조 현장에서 설비의 유지보수로 인한 입력 분포의 변화에도 출력의 분포는 변화하지 않는다.

$P_{shifted}(Y \mid X) = P_{original}(Y \mid X)$ and $P_{shifted}(X) \neq P_{original}(X)$

: Covariate shift

Domain Adaptation(DA)

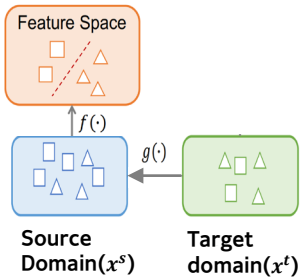


(a)두 도메인 간의 분포 차이가 있을 때, (b)Source domain으로 학습한 모형은 Target domain에서 낮은 성능을 보이게 된다.
(c)**Domain Adaptation** 방법을 적용하면 두 도메인의 입력 분포를 일치시킨 뒤, 학습한 하나의 모형으로 양 도메인의 문제를 해결할 수 있다.

Source domain은 앞선 예제에서의 분포변화 전(Original)에, Target domain은 분포변화 후(Shifted)에 대응된다.

Approach 1

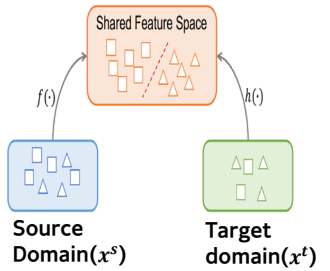
Domain Transformation



- Target domain과 Source domain의 입력 분포를 직접 일치시킴
- Source domain으로 사전학습한 모형을 이용하여, Target domain에 대한 임무 수행
- 본 프로젝트에서는, 이 접근법을 통해 Covariate Shift를 해결함

Approach 2

Latent Space Transformation



- Target domain과 Source domain의 분포를 제 3의 공간(latent)에서 일치시킴
- Latent에서의 입력 데이터를 input으로 하는 모형을 새로 학습
- 고차원 이미지 데이터의 DA에서 선호되는 접근임

1-3. 선행 연구 분석

- **선행 연구**

Title	Data Type	Task	Transformation Type	Learning Type	비고
Domain-Adversarial Training of Neural Networks (DANN)	이미지	분류	Latent-space Transformation	Unsupervised	Available in almost any feed-forward model
Source-Free Domain Adaptation via Distribution Estimation (SFDA)	이미지	분류	Latent-space Transformation	Unsupervised/Semisupervised	No need to use source data
d-SNE: Domain Adaptation using Stochastic Neighborhood Embedding (*d*-SNE)	이미지	분류	Latent-space Transformation	Supervised	Extensive to semi-supervised
DARE-GRAM : Unsupervised Domain Adaptation Regression by Aligning Inverse Gram Matrices (DAREGRAM)	이미지	회귀	Latent-space Transformation	Unsupervised	Not sensitive to hyperparameters
Multi-domain adaptation for regression under conditional distribution shift (DARC)	정형	회귀	Latent-space Transformation	Supervised	Multi-Domain

정형 데이터를 대상으로 Regression Task를 다루는 Original Space Transformation 기법에 대한 최근 연구 없음

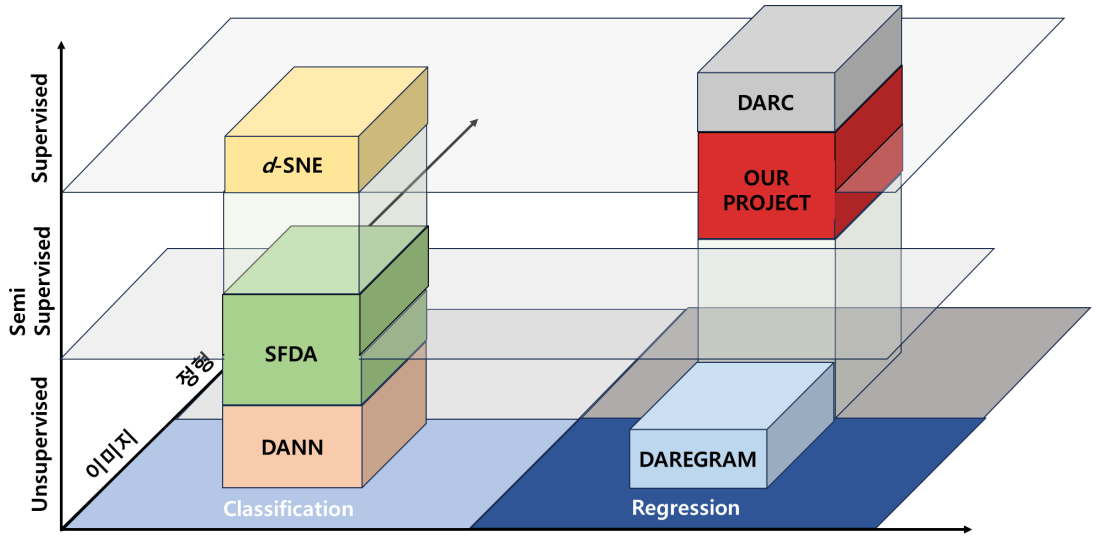
- **개선점**

정형 데이터 최적화
- 극소량의 정형 데이터에 최적화

Original Space Transformation
- 기존 회귀 모형 재사용 가능

최적 회귀 모형 추정
- Domain Adaptation에 필요한 시간과 비용 절감

- **프로젝트 포지셔닝**



2-1. 문제 정의

- **문제 정의**

Task Type

- 정형 데이터를 활용하는 제조 공정을 대상으로 함
- 공정 결과를 예측하는 회귀 과업 진행

Available Resource

- 많은 양의 Source Data를 보유한 상황
- 이를 기반으로 사전 학습된 모델 보유

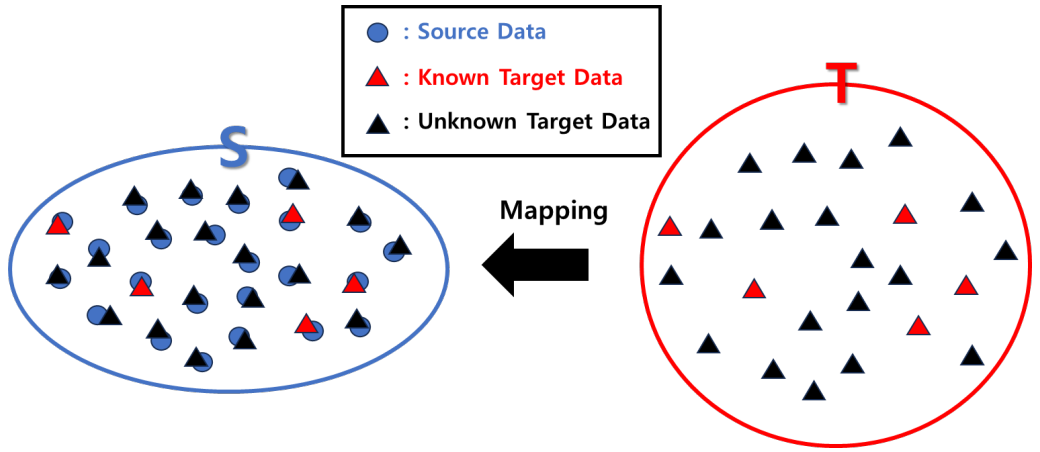
Constraint

- 예방 정비, 시설 노후화 등의 이유로 수집되는 Data에 Covariate Shift 발생
- Shift 이후 Data 부족으로 모델 재학습 난항

Main Idea

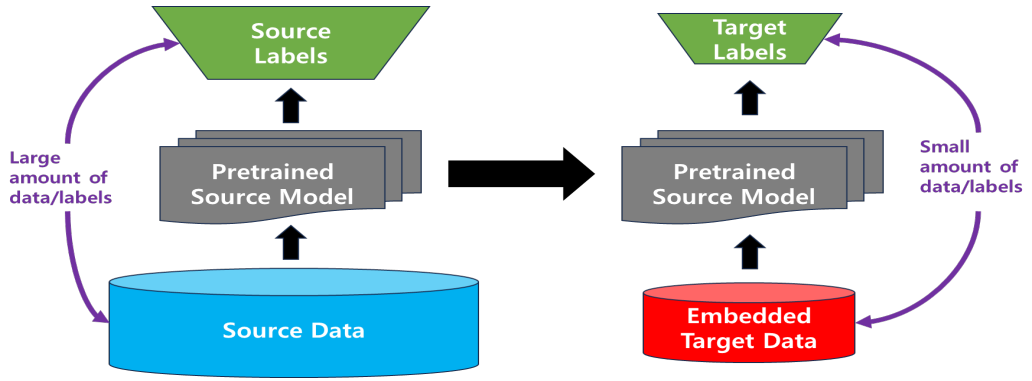
- 원활한 공정 흐름을 위해 **기존 모델 재활용**

How?



- Target Domain data를 Source Domain에 mapping시키면 Source data로 사전 학습된 모델을 재활용할 수 있음.

Use Case

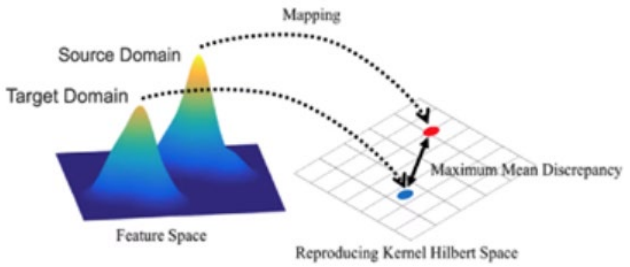


2-2. 손실함수 제안

- x^s/x^t : Source / Target input
- y^s/y^t : Source / Target output
- $f^s(x^t)$: Embedded Target on Source Domain
- n^s/n^t : Source / Target Batch Size

- **Loss 제안**

Maximum Mean Discrepancy (MMD) Loss

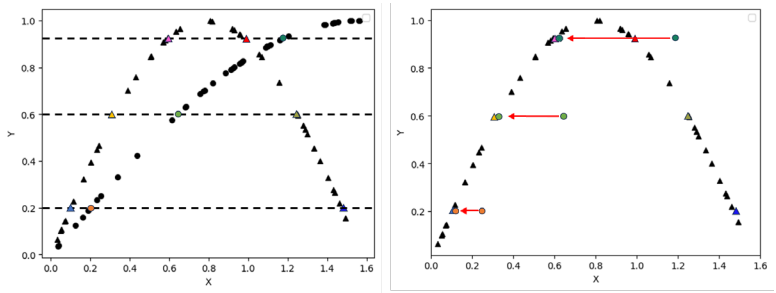


- Source Domain과 Target Domain의 입력 분포 일치

$$L_{MMD} = \max_{m}\left(\left\|\frac{1}{n^s}\sum_{j=1}^{n^s}\Phi_m(x_i^s) - \frac{1}{n^t}\sum_{j=1}^{n^t}\Phi_m(x_j^t)\right\|^2\right)$$

- M 개의 Embedding 함수 Φ_m 중 두 분포 간의 거리가 최대가 되도록하는 Φ_m 선정
- 선정된 Φ_m 을 통해 산출한 거리가 최소가 되도록 Loss update

Neighborhood Similarity (NS) Loss



- 출력값을 기반으로 각 Data point 1대1 대응

$$L_{ns} = \frac{1}{n}\sum_{i=1}^{n}\left|x_{\hat{j}}^s - f^s(x_i^t)\right|_2$$

$$(for\ \hat{j} = \arg\min_j \left|x_j^s - f^s(x_i^t)\right|_2 + \left|y_j^s - y_i^t\right|)$$

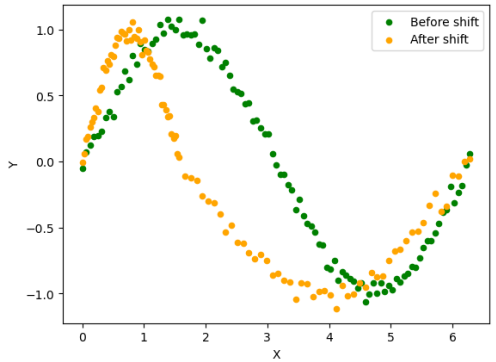
$$\{x_1^s, x_2^s, \cdots, x_k^s \,|\, |y_1^s - y_i^t| < \cdots < |y_k^s - y_i^t|\}(hyperparameter\ k)$$

- 임의의 Target data point의 y 값이 y^t 라고 할 때 그와 비슷한 y^s 값을 갖는 Source data point와 가깝게 위치

최종 Loss Function $\rightarrow \alpha \cdot L_{MMD} + \beta \cdot L_{NS}\ \ (hyperparameter\ \alpha, \beta)$

3-1. Toy data 실험

Dataset



$$\mathbf{Y} = \boldsymbol{f}(\boldsymbol{X}) + \boldsymbol{\varepsilon}$$

- $f : True\ function$
- $\varepsilon \sim N(0, 0.05)$

[분포 변화 이전]

- $P(X^s) = Unif\,(0, 2\pi)$
- $P(Y^s|X^s) = \mathrm{N}(\sin(X^s), 0.05)$

[분포 변화 이후]

- $P(X^t) = \frac{1}{2}Unif\left(0, \frac{1}{2}\pi\right) + \frac{1}{2}Unif\left(\frac{1}{2}\pi, 2\pi\right)$
- $P(Y^t|X^t) = \begin{cases} \mathrm{N}(\sin(2X^t), 0.05) & X^t \in [0, \frac{1}{2}\pi] \\ \mathrm{N}(\sin\left(\frac{2}{3}X^t + \frac{2}{3}\pi\right), 0.05) & X^t \in [\frac{1}{2}\pi, 2\pi] \end{cases}$

OURS

[모델 아키텍처]

- MLP Mapper[Target → Source] ➔ Pretrained MLP

[사전 설정값]

- $(batch\ size)\ n^s = n^t$
- $(number\ of\ Embedding\ function)\ M = 10$
- $(hyperparameter)\ \alpha = \beta = 1\ \ \&\ \ k = 3$
- $(Pretrained\ MLP\ performance)\ Source\ MSE = 0.00128$

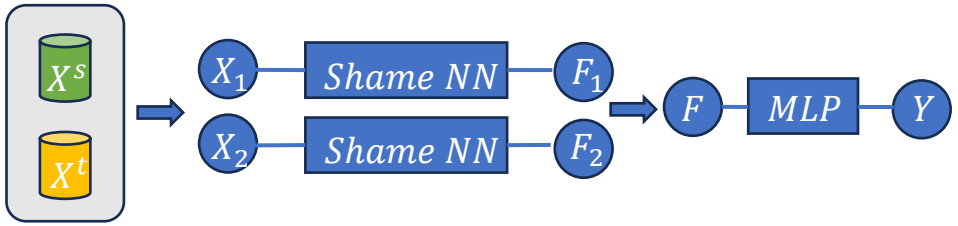
Comparison

[MLP]

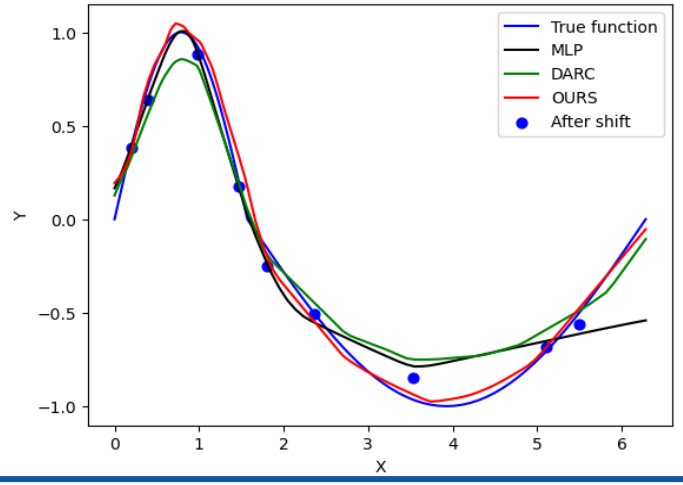
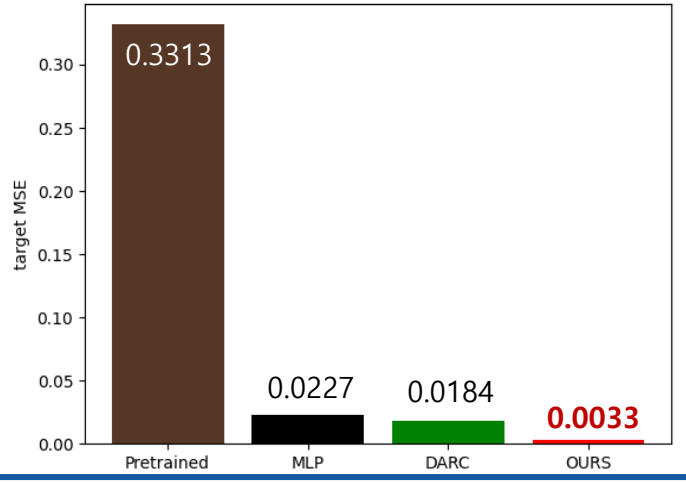
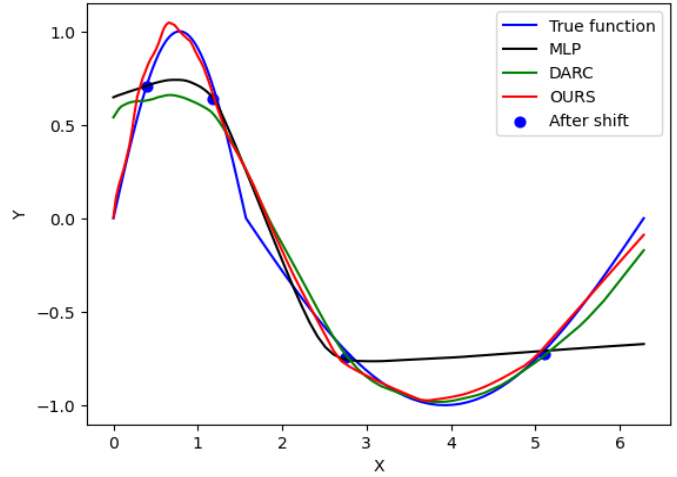
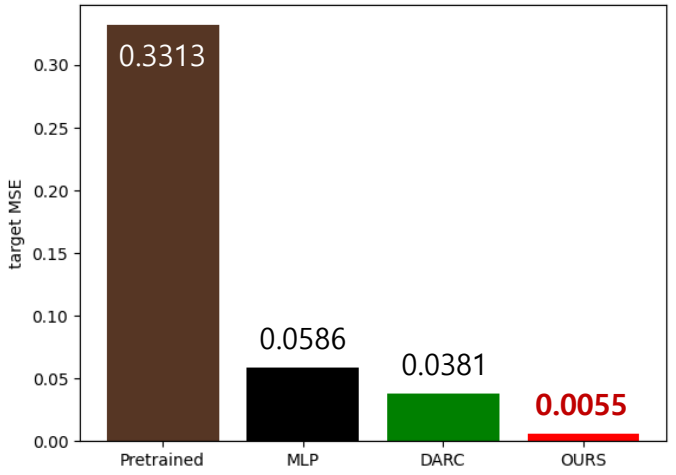
- Target Data → MLP model

[DARC]

- Source/Target Data → Shame NN → MLP model



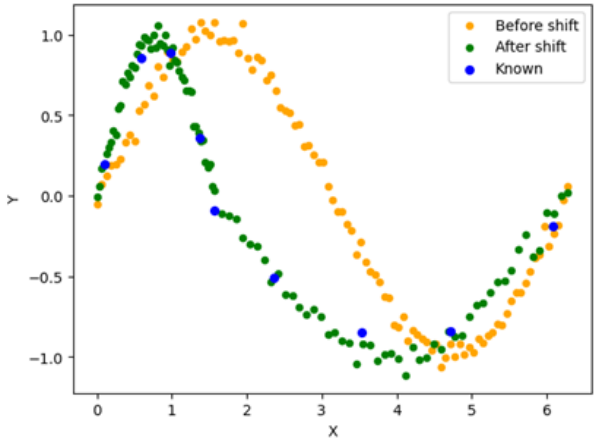
3-1. Toy data 실험 결과

데이터 수	분포 변화 이후의 데이터에 적합시킨 회귀 모형	회귀 모형 별 성능 비교
9 shot		
4 shot		

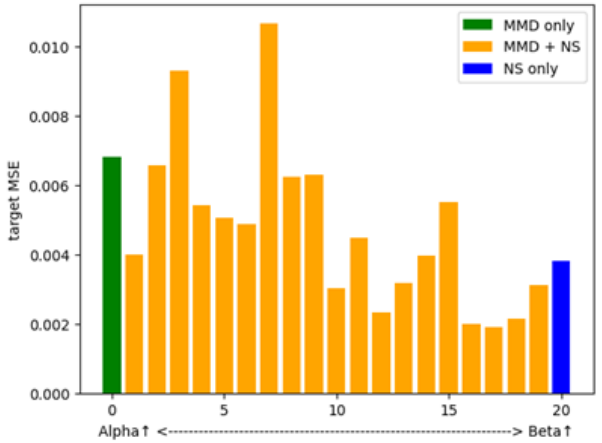
3-2. Hyperparameter Sensitivity (1)

Unbiased Data

관측된 target data가 모집단의 분포를 따르는 경우



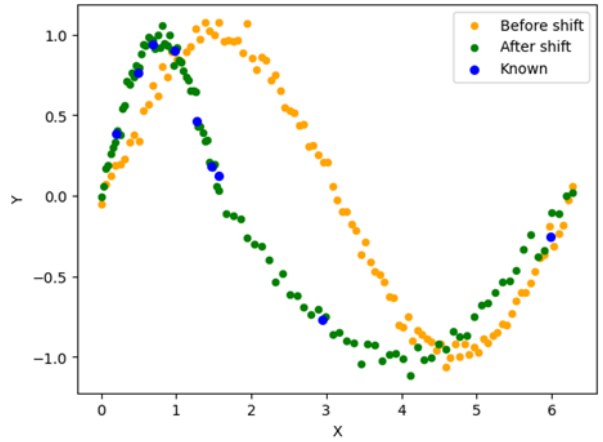
(a) 데이터 분포



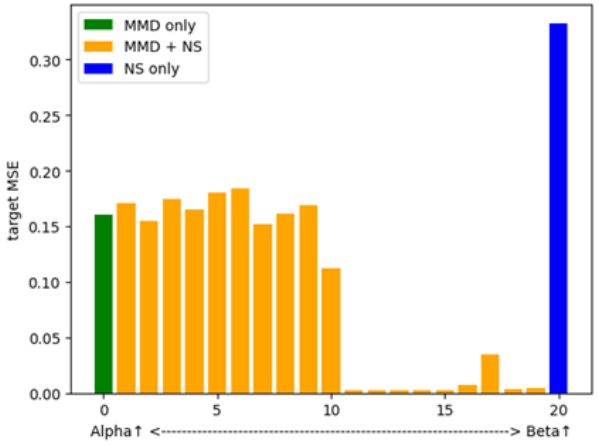
(b) Hyperparameter α, β sensitivity

Biased Data

관측된 target data가 편향되어 있는 경우



(c) 데이터 분포



(d) Hyperparameter α, β sensitivity

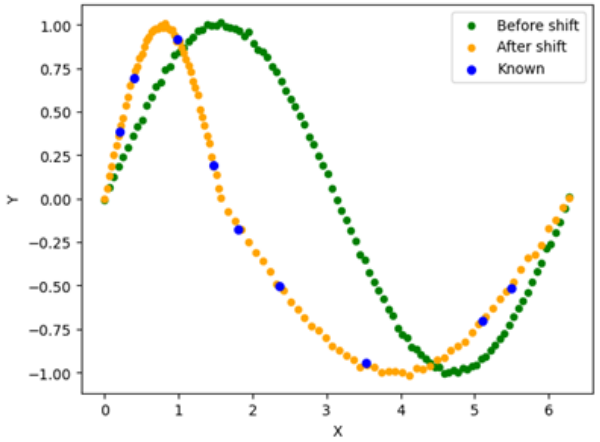
결론

1. MMD Loss와 NS Loss를 함께 minimize하는 방향으로 학습해야 좋은 성능을 보인다.
2. NS Loss에 가중치를 줄수록(β가 클수록) 편향된 데이터에 강건한 결과를 보인다.

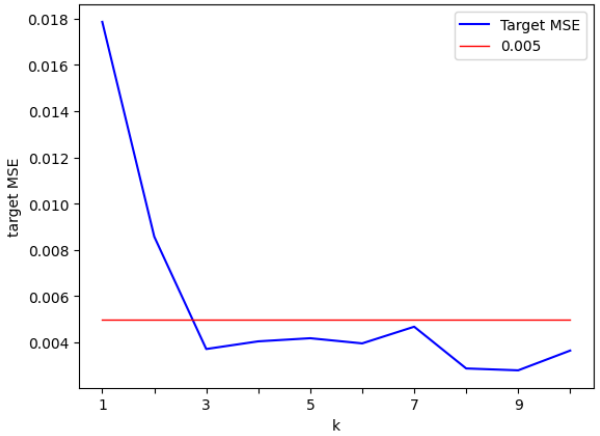
3-2. Hyperparameter Sensitivity (2)

$\epsilon \sim N(0, 0.01)$

관측된 y의 분산이 작은 경우



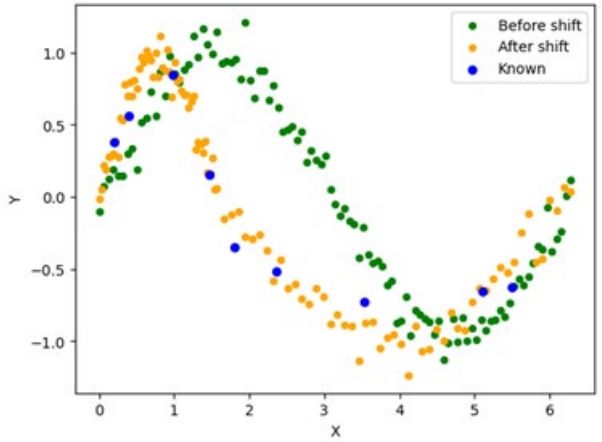
(a) 데이터 분포



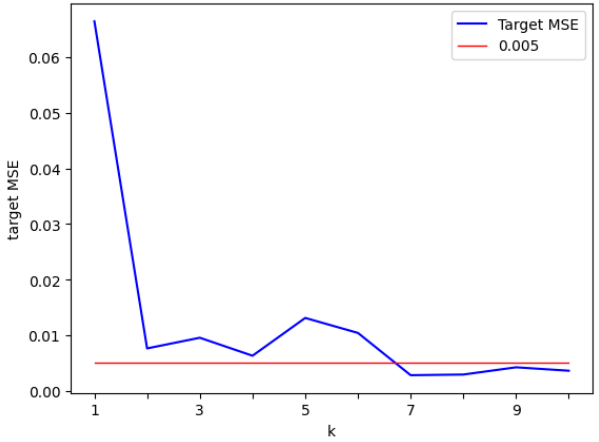
(b) Hyperparameter k sensitivity

$\epsilon \sim N(0, 0.1)$

관측된 y의 분산이 큰 경우



(c) 데이터 분포



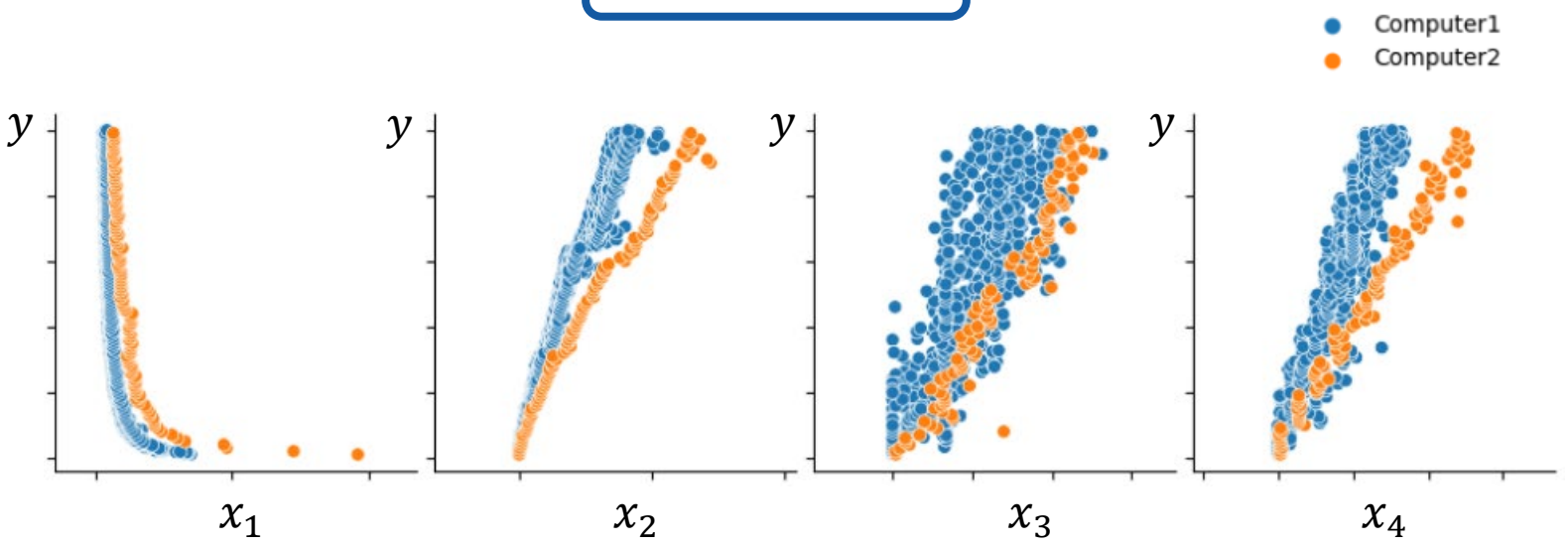
(d) Hyperparameter k sensitivity

결론

k (이웃 후보 개수) 값이 클 수록 y의 분산에 강건한 결과를 보인다.

3-3. 실제 데이터 실험 및 결과

실험 Setting

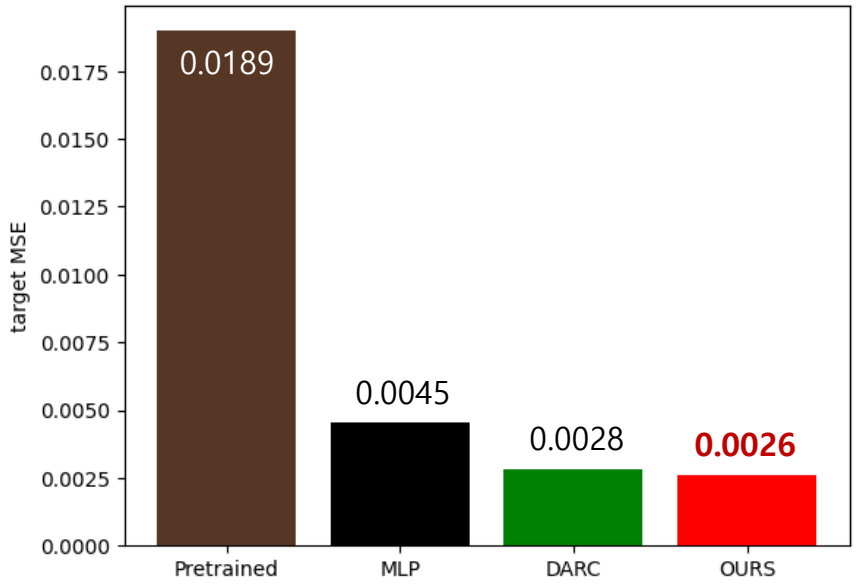


- X: 4가지 종류의 연산시간 (단위 : 초)
- Y : 연산 입력값 개수 (단위 : 백만)

- Before Shift : Computer 1 (Source)
- After Shift : Computer 2 (Target)

- $(known)\ 3\ shot$
- $(batch\ size)\ n^s = n^t = 3$
- $(number\ of\ kernel)\ M = 10$
- $(hyperparameter)\ \alpha = \beta = 1\ \&\ k = 3$
- $(Pretrained\ MLP)\ Source\ MSE = 0.00042$

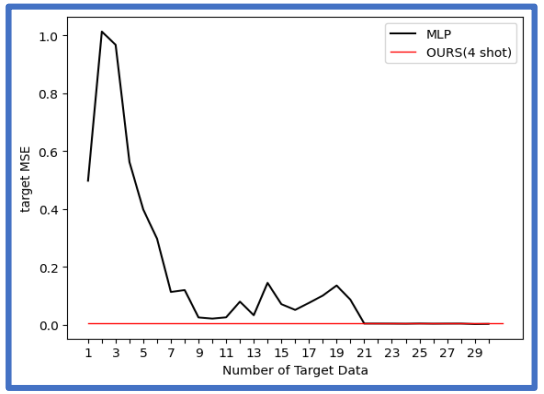
회귀 모형 별 성능 비교

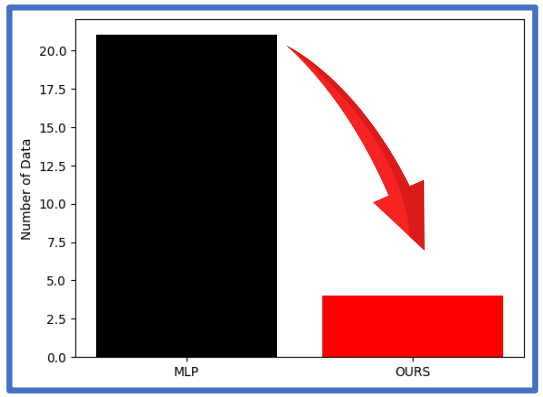


4-1. 프로젝트 기대효과

핵심가치

경제성





MLP 모델로 동일한 성능을 보이기 위해선
5배 이상의 Target Data가 필요

보존성

기존 모형을
재활용하여
정보 손실 없이
새로운 데이터에
사전 지식 반영

강건성

하이퍼파라미터에
강건하여
현장 조업자들의
작업난이도 감소

유연성

정형데이터를
다루는 모든
문제에 적용 가능

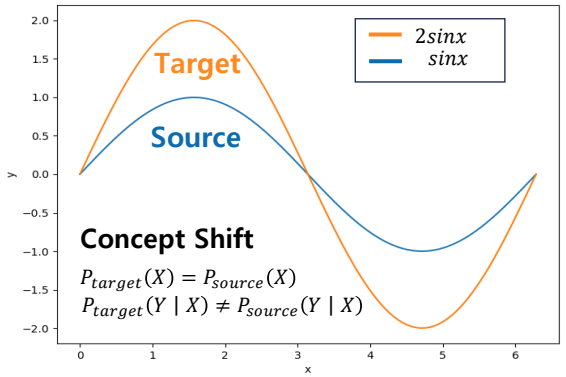
다양한 실무 환경에 적용 가능

4-2. 향후 개선점 토의

실제 공정 데이터 적용

- 본 프로젝트에서는 실제 공정 데이터에 대한 실험을 진행하지 못하였음.
- 실제 공정 데이터에 대한 검증을 통하여, 제안 방법의 실효성에 대한 구체화가 필요함.

Concept Shift로의 확장



- Concept Shift란, 입력의 분포가 동일할 때, 출력의 분포가 상이한 분포 변화를 일컫는다.
- 본 프로젝트의 Covariate Shift에 더하여 Concept Shift까지 확장할 수 있다면, 실제 현장의 문제 해결 범용성을 확대할 수 있을 것임.

데이터 분포 변화 감지 기능 추가

- 본 프로젝트에서는 도메인 지식을 통해, 데이터 분포 변화의 시점을 파악하였다고 가정함.
- 이상 탐지 등의 방법론 적용을 통해, 데이터 분포의 변화를 인식하고, 그 후에 제안 방법을 사용하여 분포 변화에 대응하는 End to End 시스템을 구축하면 그 활용가치가 클 것임.

"데이터 분포의 **변화에 대응**한다는 것은 AI 이론과 제조 현장 간의 괴리를 허물기 위해 내딛어야 할 **필연적인 첫걸음**입니다."