

Pseudo Sample Generation for Hyperparameter Selection of One Class SVM

지능데이터시스템 연구실(IDSL)

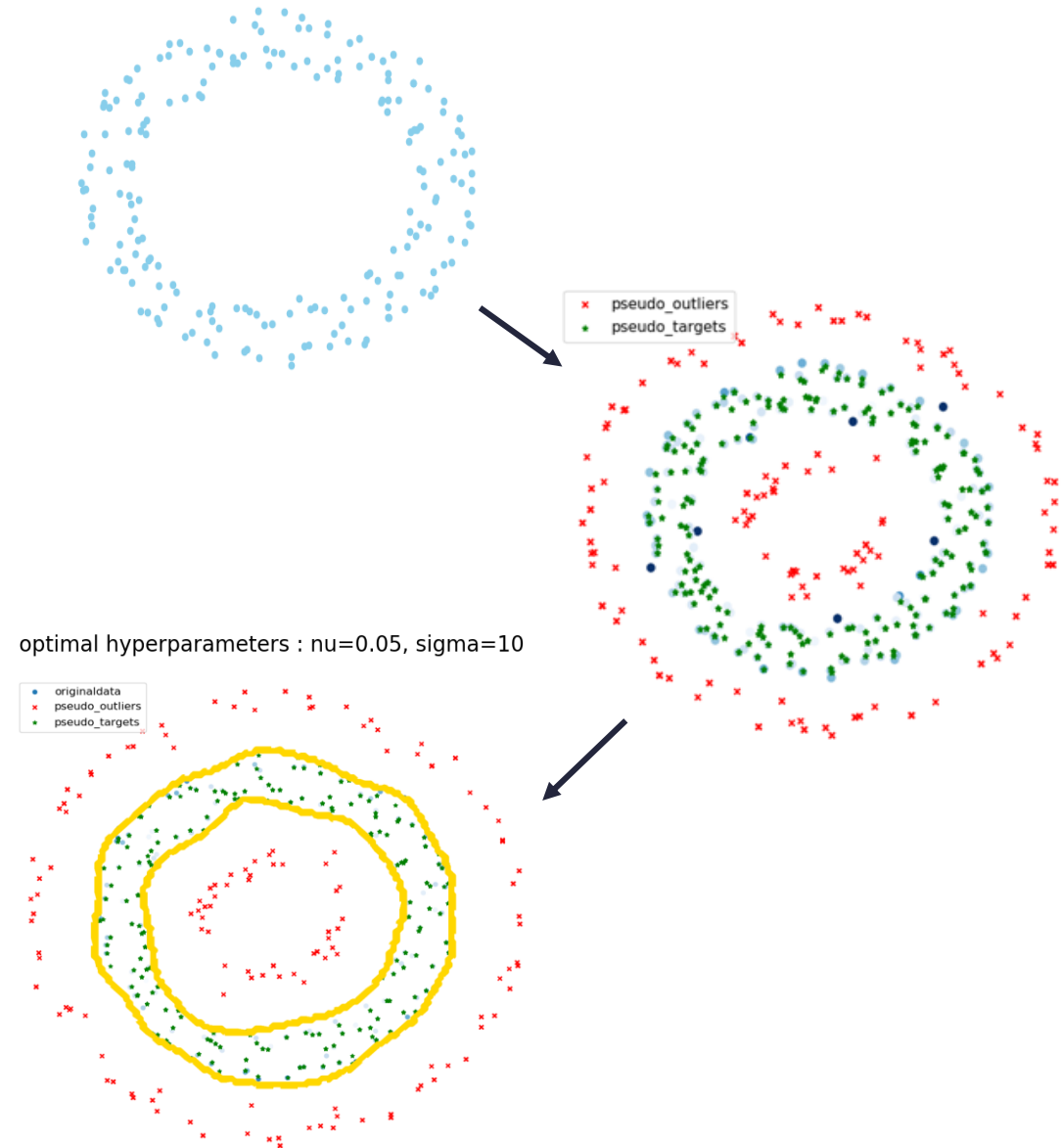
우 필 원

https://github.com/woopal/PSG_for_OCSVM

- ❖ 본 알고리즘은 이상치 탐지를 위한 One Class SVM(OCSVM)의 성능 개선을 위해 고안 되었으며 한양대학교 산학협력 프로젝트에 적용 될 예정입니다.

구현, 실험 코드(Python3.7)
https://github.com/woopal/PSG_for_OCSVM

1. 최종 목적 : OCSVM Hyperparameter (ν, σ) selection method의 제시
2. 핵심 수단 : 학습 데이터의 기하학적인 특성으로부터 가상의 샘플(pseudo-sample)을 생성하여 (ν, σ) 를 위한 validation set으로 활용
3. Contribution : 기존의 방안보다 기하학적인 특성을 더 잘 반영할 수 있는 pseudo-sample 생성 방안을 제시
4. Result : 15개의 UCI/LIBSVM dataset에 대해 비교실험 결과 성능의 개선이 이루어짐을 확인



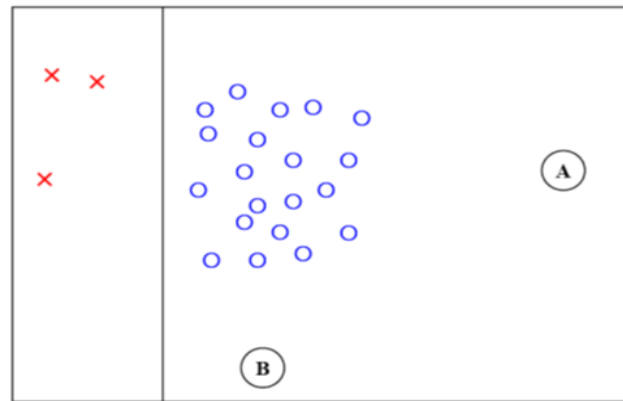
Index

1. Introduction : One Class SVM
2. Main problem : Hyperparameter Selection
3. Solution : Pseudo Sample Generation
4. Main idea : New method to generate Pseudo Samples
5. Experiments
6. Conclusion

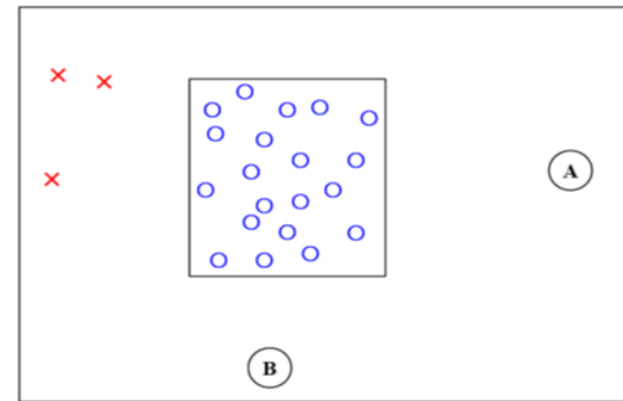
1. Introduction : One Class SVM

Novelty detection(이상치 탐지)

- ✓ 다수의 정상 데이터만 존재하고 이상치의 데이터가 없거나 소수이며, 습득하기 힘든 상황
- ✓ 분류(classification)로써 접근이 불가능하여 정상 데이터에 대한 decision boundary를 형성해주는 Unsupervised learning 기법
- ✓ 기존의 정상 분포와 다른 이상 데이터의 발생을 탐지하기 위함



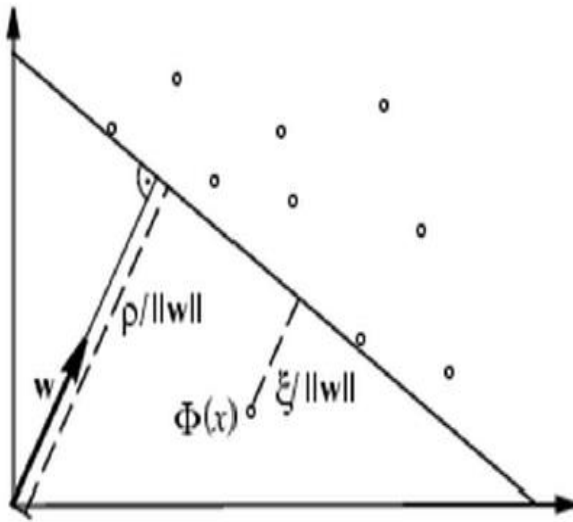
Binary classification



Novelty Detection

One Class SVM (OC-SVM)

- ✓ 이상치 탐지를 위해 널리 쓰이는 SVM기반의 알고리즘(Scholkopf *et al.*, 2000)
- ✓ 원점(Origin)으로부터 대부분의 데이터를 분리해내는 Hyperplane $\langle \mathbf{x}, \Phi(\mathbf{x}) \rangle = \rho$ 의 학습
- ✓ Gaussian Kernel trick를 통해 비 선형의 decision boundary 학습이 가능함



Primal Quadratic Program

$$\min_{\mathbf{w}, \xi, \rho} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu m} \sum_{i=1}^m \xi_i - \rho,$$

subject to $\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle \geq \rho - \xi_i, \quad \xi_i \geq 0$

Dual Quadratic Program

$$\min_{\alpha} \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j),$$

subject to $0 \leq \alpha_i \leq \frac{1}{\nu m}, \quad \sum_{i=1}^m \alpha_i = 1.$



Lagrange multiplier,
KKT conditions

Adapted from Lecture note of Arcolano, Rudoy (Harvard University)

Challenge of One Class SVM

2개의 Hyperparameter 설정이 사용자의 몫으로 남아있음

1. σ (gaussian kernel bandwidth)

2. ν (regularization coef) $\in [0,1]$

Dual Quadratic Program

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(x_i, x_j), \\ \text{subject to} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu m}, \quad \sum_{i=1}^m \alpha_i = 1. \end{aligned}$$

(2)

Gaussian kernels:

$$K(x_1, x_2) = \exp \left\{ -\frac{\|x_1 - x_2\|_2^2}{2\sigma^2} \right\}, \quad \sigma \neq 0$$

(1)

2. Main problem : Hyperparameter Selection

2. Main problem : Hyperparameter Selection

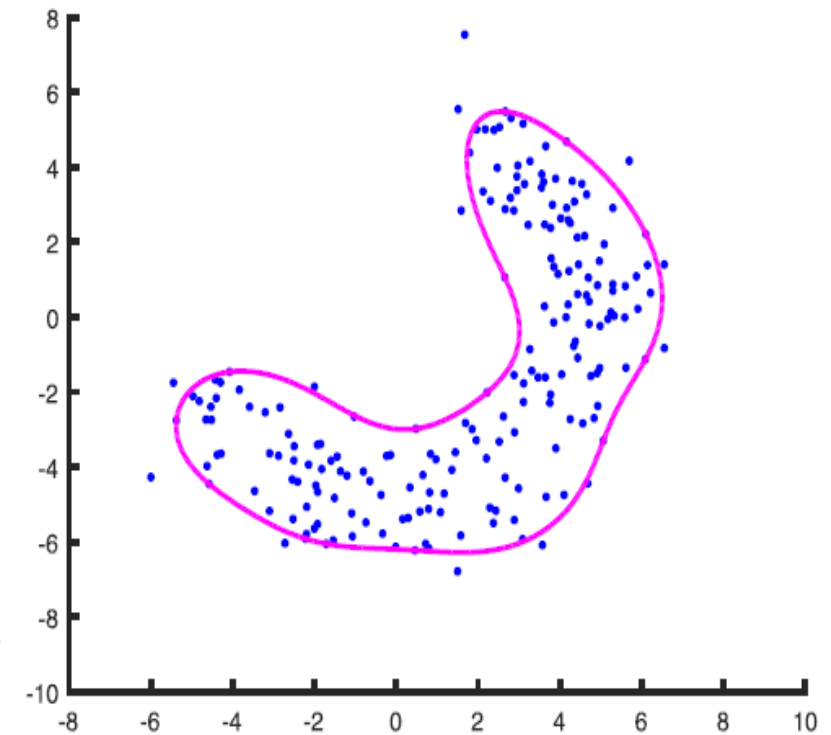
Hyperparameters of OC-SVM

1. ν (regularization coef) $\in [0,1]$

Training data 중 decision boundary 바깥으로 최대한 허용 가능한 sample의 비율

2. σ (gaussian kernel bandwithh)

무한한 차원에서의 projection을 통해 비선형의 decision boundary 형성이 가능하게 하는
Gaussian kernel의 hyperparameter



(b) $\nu = 0.1, \sigma = \sqrt{10}$

2. Main problem : Hyperparameter Selection

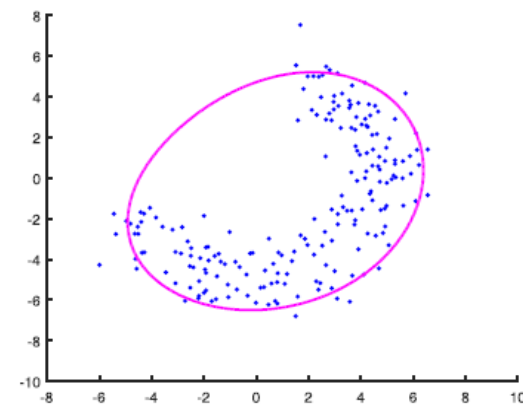
Influence of hyperparameters of OCSVM

1. ν (regularization coef)

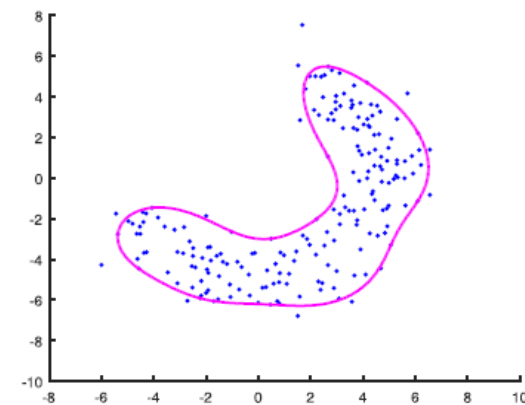
- ✓ 낮은 ν 의 설정(d)
 - 대부분의 sample을 boundary 안으로 포함
 - 지나친 generalization을 불러옴
 - 이상치 탐지 성능 저하

2. σ (gaussian kernel bandwithh)

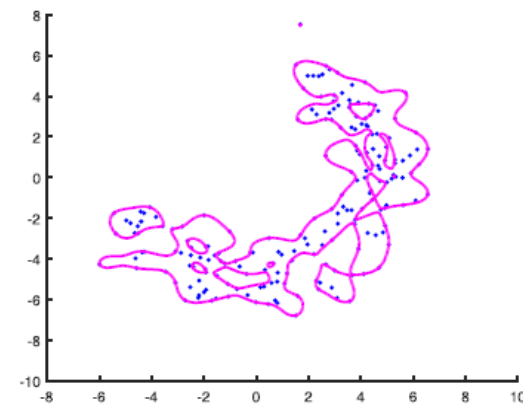
- ✓ 높은 σ 의 설정 : (a)
 - 분포의 형태를 고려하지 못하는 느슨한 decision boundary 형성
 - 이상치 탐지 성능 저하
- ✓ 낮은 σ 의 설정 : (c)
 - 과도한 Nonlinearity를 허용
 - 학습 데이터에 과적합한 decision boundary 형성
 - 정상 탐지 성능 저하



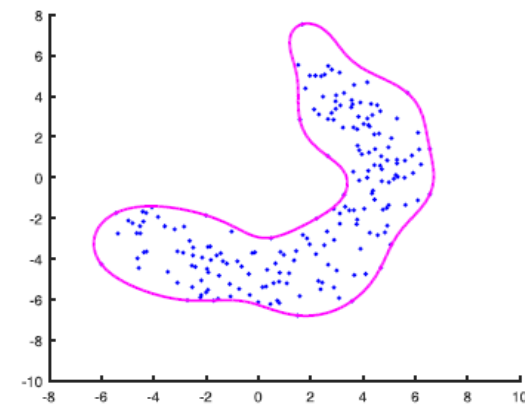
(a) $\nu = 0.1, \sigma = 10$



(b) $\nu = 0.1, \sigma = \sqrt{10}$



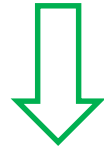
(c) $\nu = 0.1, \sigma = 1$



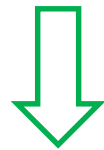
(d) $\nu = 0.01, \sigma = \sqrt{10}$

How can we set appropriate hyperparameter set?

(ν, σ) 는 OCSVM의 Hyperparameter set으로써 Decision boundary의 형성에 결정적인 영향을 미침



데이터셋의 형태를 보존하면서 Overfitting을 방지하는 최적의 (ν, σ) set을 설정 하는 방법이 있는가?



Pseudo Sample Generation

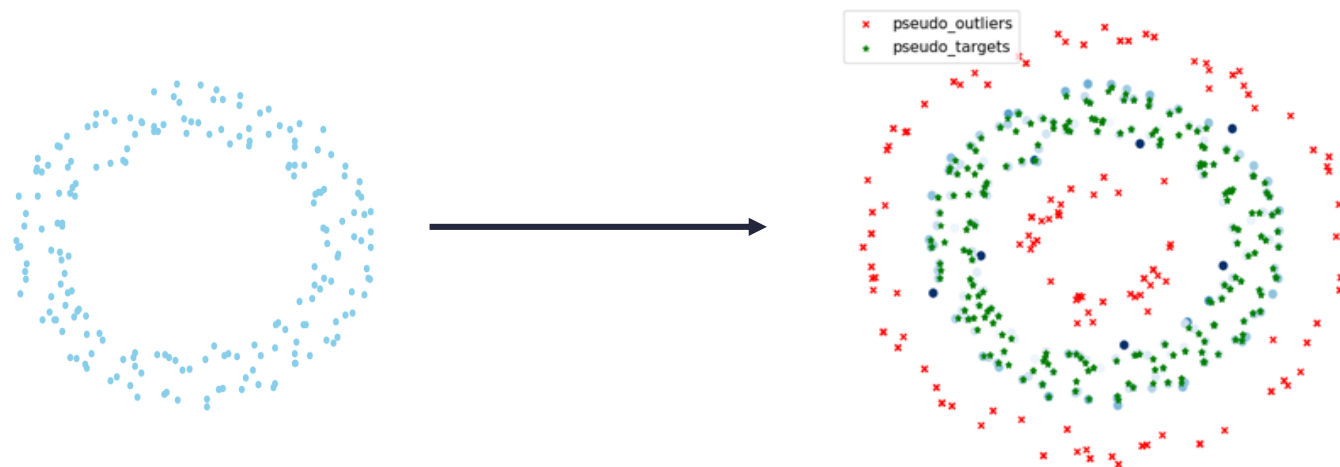
3. Solution : Pseudo Sample Generation

Pseudo Sample Generation

Training data의 분포 바깥에 Pseudo-outlier sample,

Training data의 분포 내부에 Pseudo-target sample을 생성하여

(ν, σ) 를 위한 validation set으로 사용하는 것



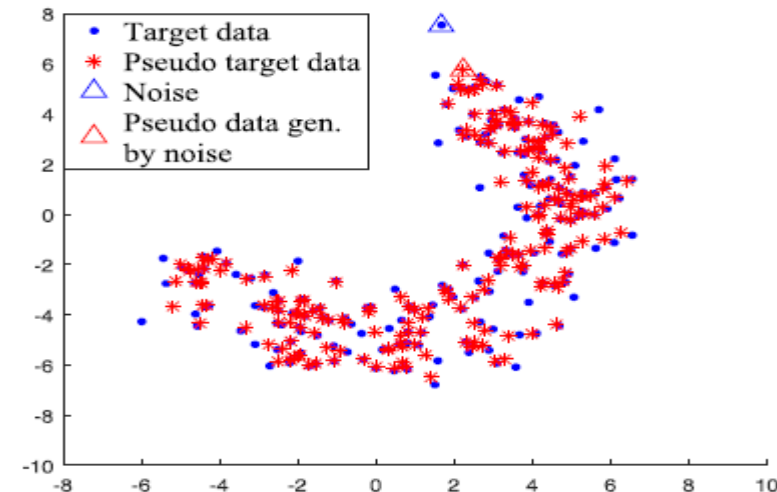
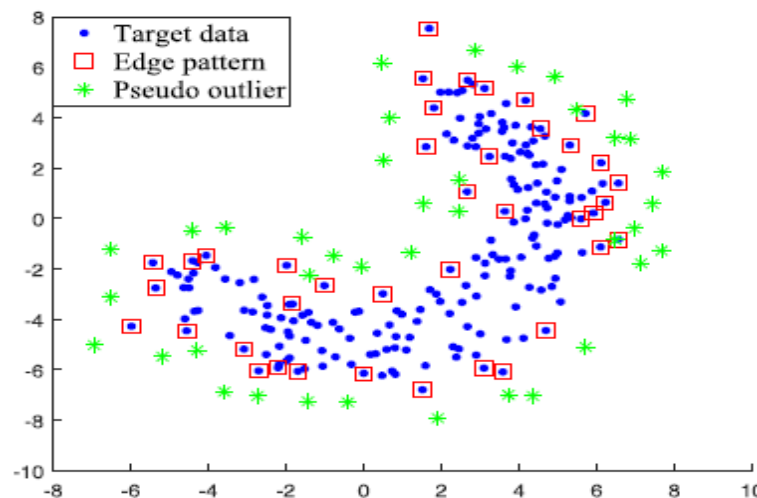
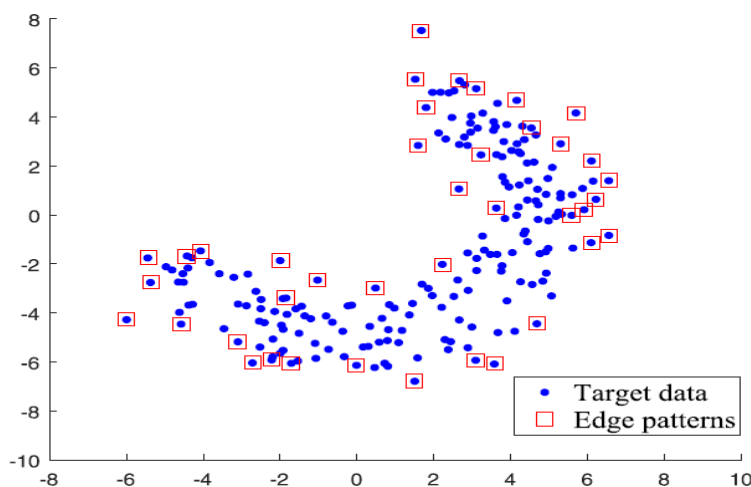
How Pseudo Samples Work?

- ✓ Pseudo-outlier는 최대한 포함하지 않고, Pseudo-target은 최대한 포함하는 decision boundary를 유도
- ✓ Pseudo-target을 포함하도록 함으로써 Decision boundary에 구멍이 생기는 등의 지나친 비선형성을 막아
정상 데이터의 분포를 보존
- ✓ Pseudo-outlier를 최대한 배제하도록 함으로써 Outlier 탐지 성능이 떨어지는 느슨한 Decision boundary를 방지

3. Solution : Pseudo Sample Generation

Existing Pseudo Sample Generation method : SDS

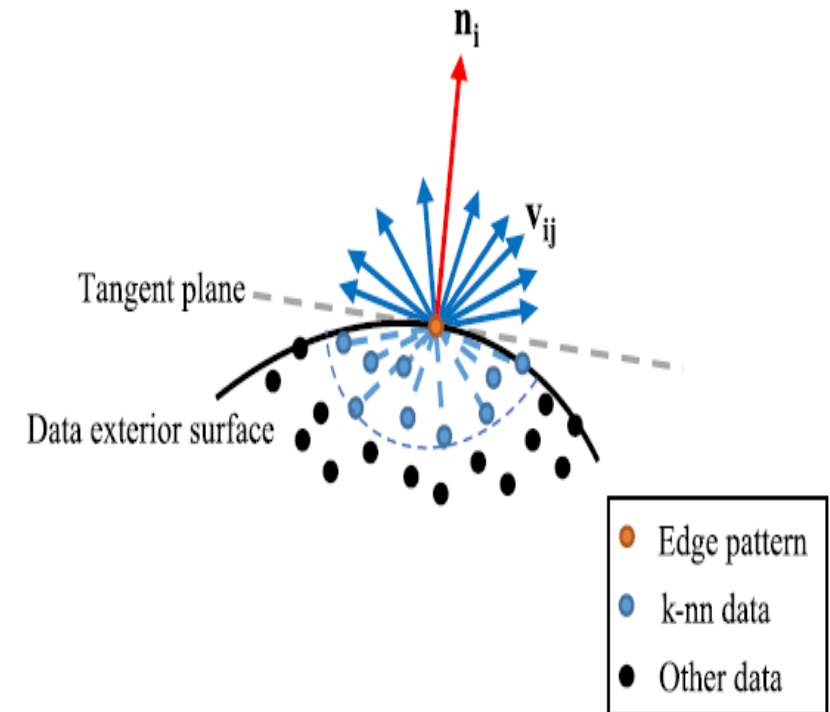
- ✓ S. Wang et al.(2018) - Self adaptive shifting(SDS) : 현재 가장 좋은 실험성능을 보이는 방법
- ✓ 데이터 분포의 외각에 위치하는 sample을 기하학적 특성을 이용하여 Edge pattern으로 판정
- ✓ Edge pattern으로 판정된 sample을 통해 Pseudo-outlier를 생성, 그 외의 sample을 통해 pseudo-target 생성



Limitations of SDS method

1) Edge pattern detection method가 갖는 문제

- ✓ SDS는 Li, Y. et al.(2011)의 방법을 차용하여 Edge pattern을 판별함
 - step 1) 판별 대상인 sample x_i 로부터, K-nearest neighbors와 sample과의 방향벡터($v_{ij}, j = 1, \dots, k$)를 k 개 구하고 합하여 분포의 surface에 대한 tangent plane(접평면)의 normal vector(n_i)를 추정
 - step 2) K-nearest neighbors 중에서 n_i 의 방향과 같은 v_{ij} 를 만드는 neighbors의 비율을 계산
 - step 3) 해당 비율이 연구자가 자체 설정한 threshold T 를 넘으면 sample x_i 를 edge pattern으로 판정 (분포의 외각에 있을 수록 대부분의 v_{ij} 와 n_i 가 같은 방향을 이루기 때문)
- ✓ 해당 방법은 K-nn을 위한 K 외에 threshold라는 새로운 hyperparameter를 연구자가 취사선택해야 하는 문제점이 있음
- ✓ 또한 그림처럼 convex한 형태가 아닌 concave한 데이터 분포의 표면에 있는 sample들은 edge로써 판별하지 못하고 넘어갈 가능성이 큼



Limitations of SDS method

2) Edge pattern을 결정적으로 구분함으로써 생기는 문제

- ✓ Edge pattern으로 판정 된 sample만 pseudo outlier generation에 쓰일 수 있음. 즉 제한적인 수의 pseudo-outlier만 생성 가능
- ✓ 외각에 위치한 sample도 Edge pattern으로 판정되지 못하고 지나치게 되면 그 주변엔 pseudo outlier generation이 이루어지지 못함
- ✓ Edge pattern으로 판정 된 sample은 pseudo target generation에 쓰이지 못함



모든 sample이 pseudo-outlier/pseudo-target의 생성 가능성을 갖도록 하자

4. Main idea : New method to generate pseudo samples

How to solve problems of SDS?

- ✓ Sample(x_i)마다 *Probability to edge*(x_i) 를 부여
- ✓ 데이터 분포의 외각(edge)에 위치 할수록 높은 Probability를 갖도록 함
- ✓ 부여된 *Probability to edge*(x_i) 와 $U \sim \text{unif}(0,1)$ 를 이용한 acceptance/rejection 을 반복하여 Pseudo outlier/pseudo target generation이 자연스레 이루어지도록 함



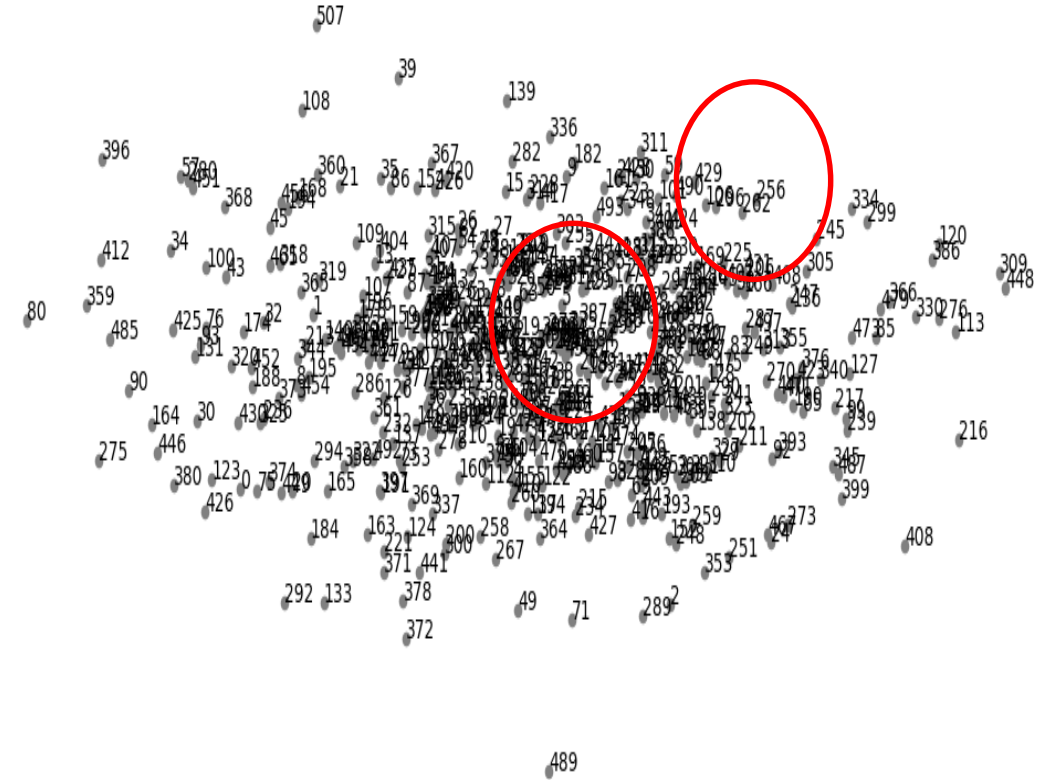
어떻게 확률을 Modeling 할 것인가?

4. Main idea : New method to generate pseudo samples

Modeling Probability for each sample

✓ Edge에 가까운 sample이 갖는 2가지 기하학적 특징을 이용

1. 분포의 Edge에 가까운 sample은, sample중심의 sphere를 생성했을 때, 분포 내부의 위치한 sample보다 상대적으로 적은 개수의 이웃이 sphere 내부로 들어올 것이다
→ feature 1(f_1): sphere 내부에 들어온 이웃의 개수. Edge에 가까울 수록 작은 값을 갖는다.
2. Edge에 가까운 sample은 sphere 내부의 이웃들이 고르게 분포하지 않고 한 방향으로 쏠려 있을 것이다.
→ feature 2(f_2): sphere 내부에 들어온 이웃들의 중점(centroid)과 sample의 거리. Edge에 가까울 수록 큰 값을 갖는다.



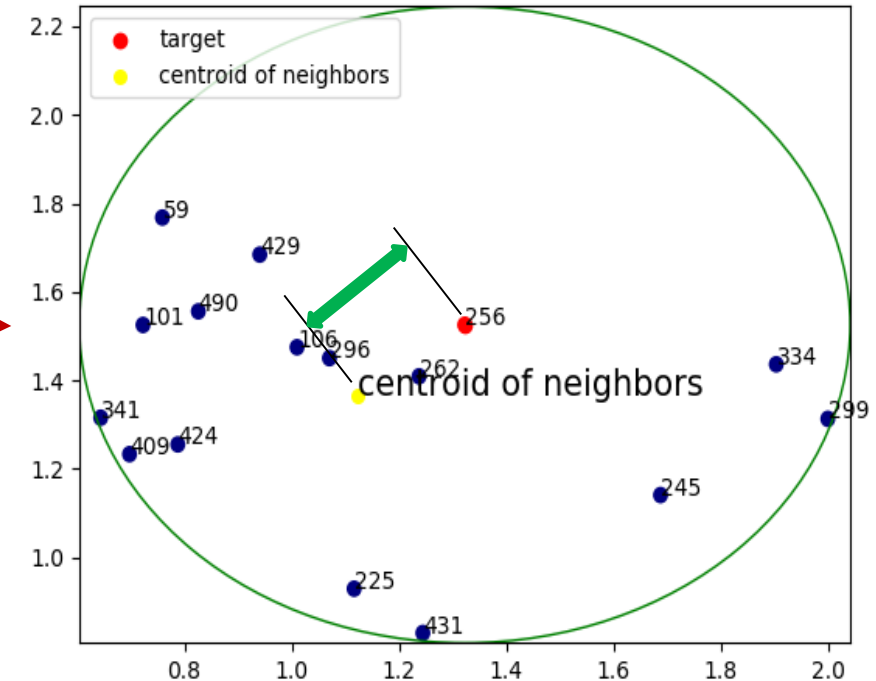
4. Main idea : New method to generate pseudo samples

Modeling Probability for each sample

Edge에 가까운 sample:

상대적으로 적은 개수의 점($f1$)이 잡히고, 상대적으로 먼 거리($f2$)를 갖는다

- ✓ $f1$: sphere 내부에 들어온 neighbor(navy point)의 갯수
- ✓ $f2$: sphere 내부의 neighbor들의 centroid와 sample과의 거리



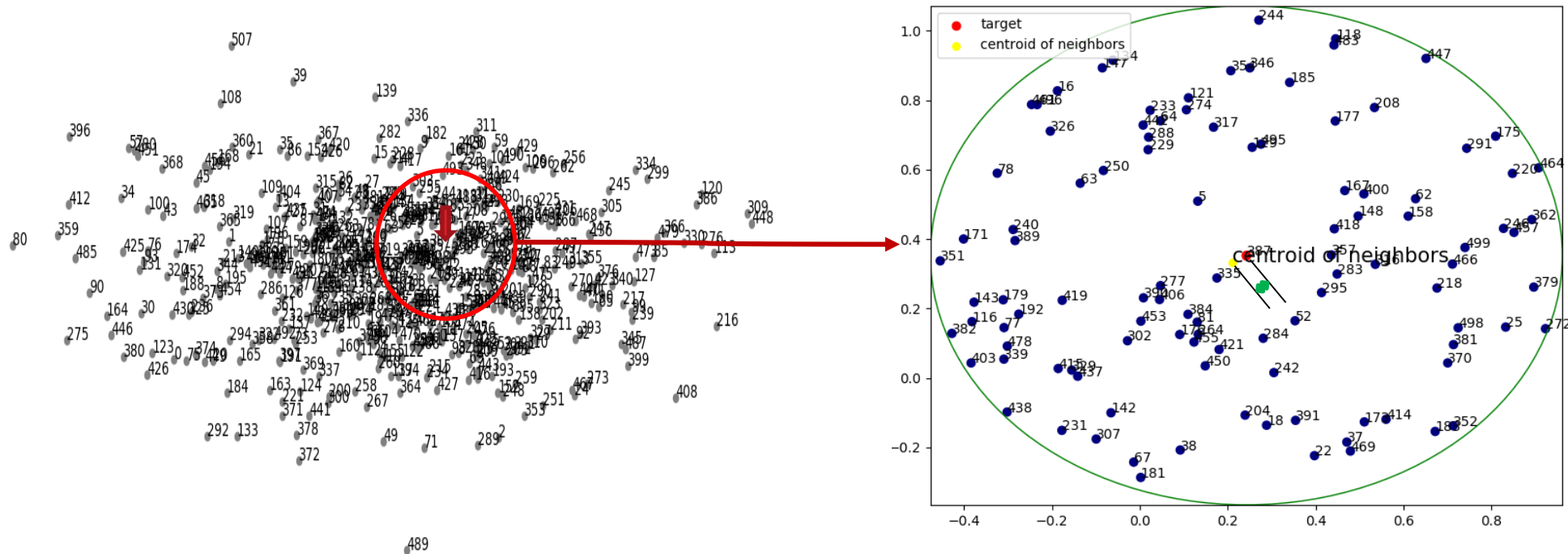
4. Main idea : New method to generate pseudo samples

Modeling Probability for each sample

Edge가 아닌, 분포의 중심에 가까운 sample:

상대적으로 많은 개수의 점($f1$)이 잡히고, 상대적으로 낮은 거리($f2$)를 갖는다

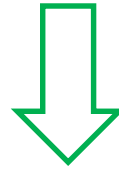
- ✓ $f1$: sphere 내부에 들어온 neighbor(navy point)의 갯수
- ✓ $f2$: sphere 내부의 neighbor들의 centroid와 sample과의 거리



4. Main idea : New method to generate pseudo samples

Modeling Probability for each sample

- ✓ Probability to edge= $F2 - F1$ 로 모델링한다면, Edge에 가까운 sample이 상대적으로 높은 Probability를 갖는다.



$$Probability\ to\ edge(x_i) = \frac{(f2_i - f1_i) - \min(f2 - f1)}{\max(f2 - f1) - \min(f2 - f1)} \in [0, 1]$$

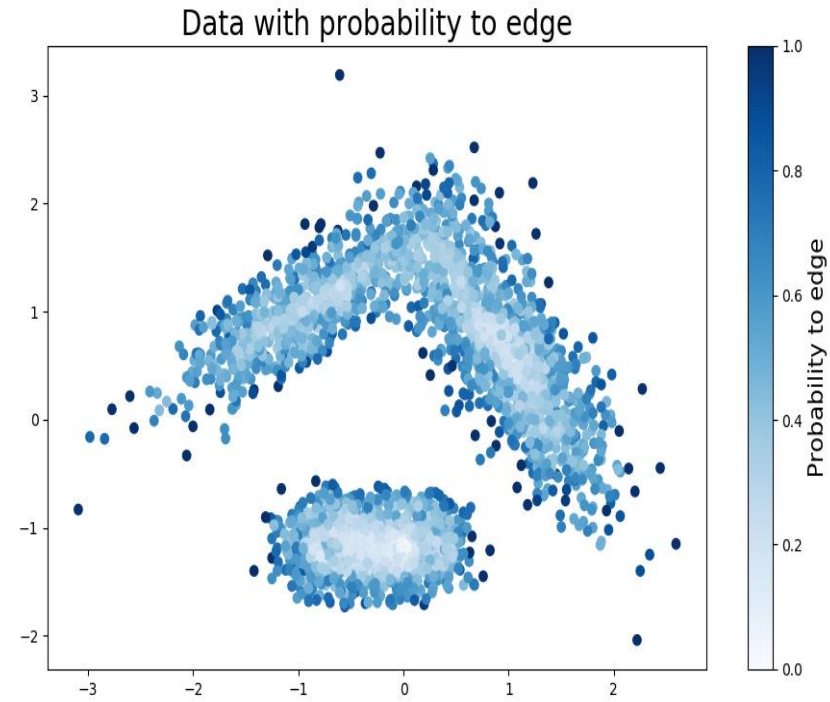
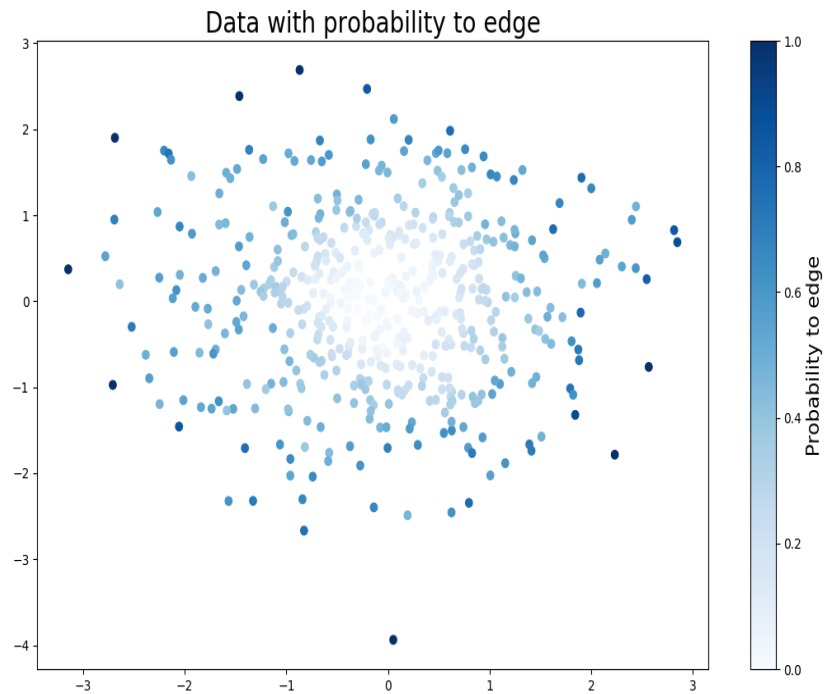
$$i = 1, \dots, N$$

$$N = \text{number of samples}$$

4. Main idea : New method to generate pseudo samples

Modeling Probability for each sample

- ✓ 그라데이션을 이용한 $Probability\ to\ edge(x_i)$ 의 시각화 결과
- ✓ 분포 내부의 sample보다 외각의 sample이 더 높은 $Probability\ to\ edge(x_i)$ 를 갖는다.



4. Main idea : New method to generate pseudo samples

Generating Pseudo samples with $U \sim \text{unif}(0,1)$

for x_i in $\{x_1, x_2, \dots, x_N\}$ **do**

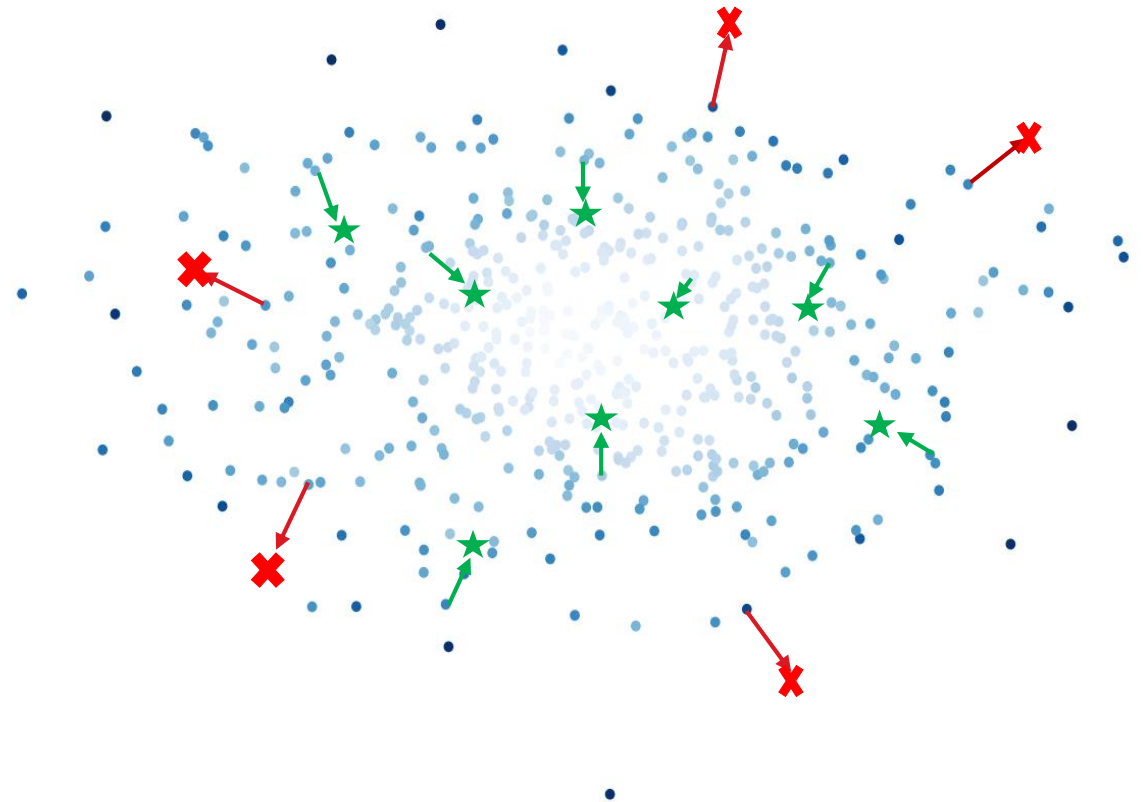
1. Generate $U \sim \text{unif}(0,1)$

2. **if** $U \sim \text{unif}(0,1) < \text{Probability to edge}(x_i)$:

generate pseudo – outlier(x_i)

else:

generate pseudo – target(x_i)



4. Main idea : New method to generate pseudo samples

Direction of Pseudo samples

pseudo – sample(x_i) 생성 방향의 key :
data density gradient

- ✓ x_i : training sample
- ✓ $x_{neighbor}$: x_i 의 sphere 내부에 들어온 sample

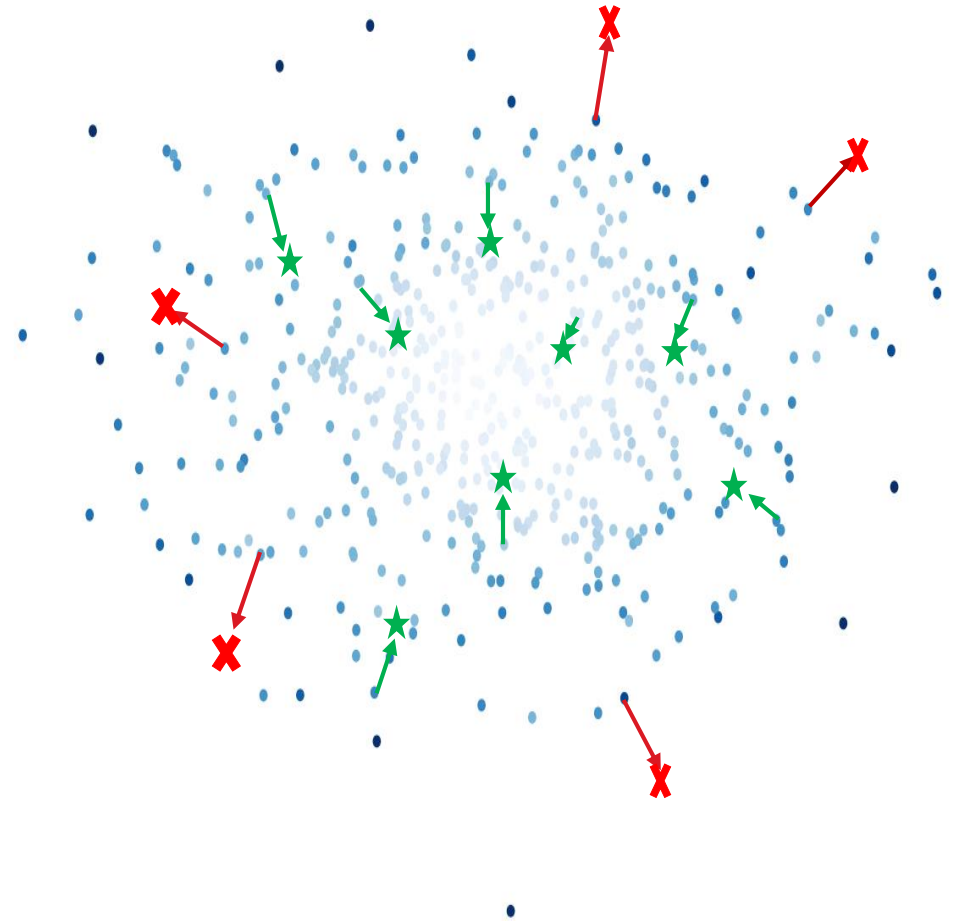
$$n_i = \sum_{neighbor} x_i - x_{neighbor}$$

\approx direction of data density gradient

Following method in K. Fukunaga(1990)



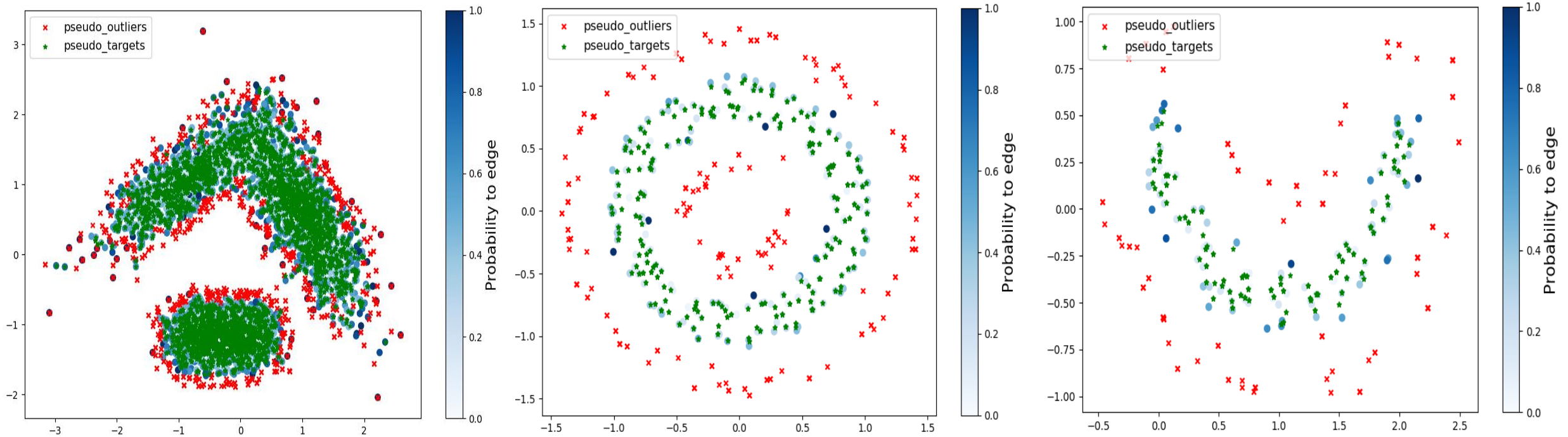
- ✓ *pseudo – target*(x_i) : Density-gradient 의 방향 n_i
- ✓ *pseudo – outlier*(x_i) : Density-gradient 의 역 방향 $-n_i$



4. Main idea : New method to generate pseudo samples

Pseudo samples on 2D dataset

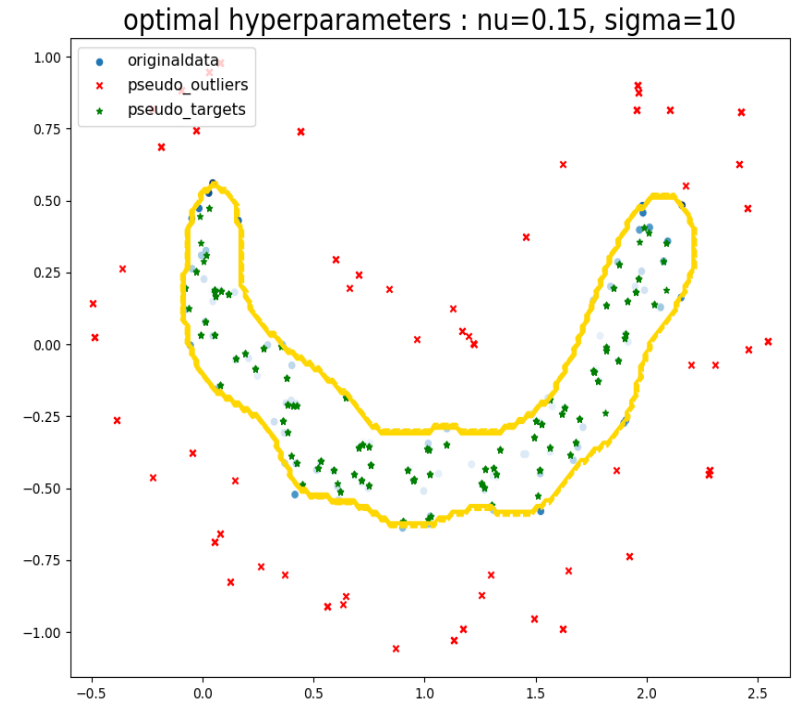
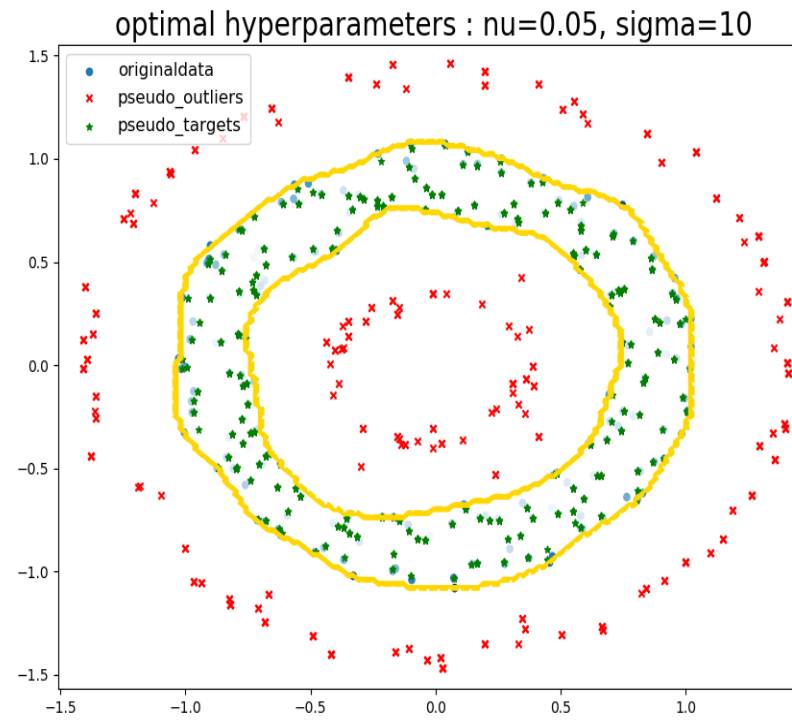
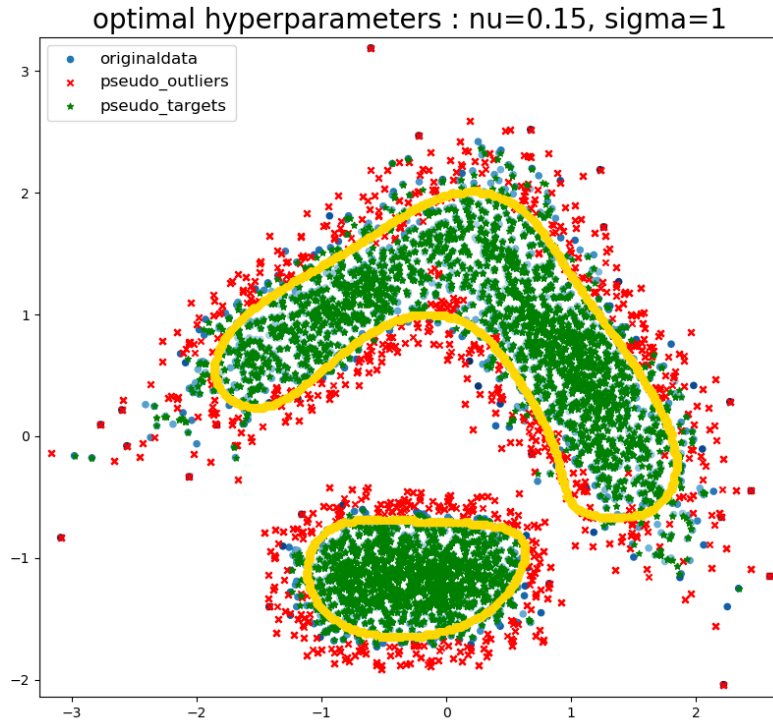
- ✓ Synthetic 2-D dataset에 대한 Pseudo sample 생성 결과
- ✓ Ring, moon 형태의 Convex하지 않은 곡면에서도 고르게 생성 되는 pseudo-outliers
- ✓ 분포를 보존할 수 있도록 생성된 pseudo-targets



4. Main idea : New method to generate pseudo samples

Search optimal (ν, σ) with Pseudo samples

- ✓ $\sigma \in [10^{-4}, 10^{-4}, \dots, 10^4]$, $\nu \in [0.01, 0.05, 0.1]$: SDS와 같은 setting
- ✓ (ν, σ) 의 모든 combination에 걸쳐 training data를 통한 OCSVM 학습
- ✓ Pseudo outlier/Pseudo target에 대한 classification error가 가장 낮은 $(\nu_{opt}, \sigma_{opt})$ 선택



Sphere 생성을 위한 radius 설정

- ✓ $f1(\text{sphere 내부에 잡힌 sample의 수})$ 의 정렬이 $\{x_1, x_2, \dots, x_N\}$ 에 대해 고르게 이루어지도록
 - ✓ $K = 5 \ln N$ ($N = \text{number of data}$) from Li, Y. et al.(2011)
- ✓ $\text{radius}(x_i)$: x_i 를 중심으로 k 개의 nearest neighbor을 포함 할 수 있는 sphere의 최소 반지름
 - ✓ $\text{radius}_{model} = \sum_{x_i} \text{radius}(x_i) / N$

5. Experiments

Results on benchmark datasets

- ✓ UCI Machine Learning Repository/LIBSVM webpage의 15개의 dataset을 사용
- ✓ 4~60의 다양한 dimension에 대하여, 다른 8개의 OCSVM hyperparameter selection method와 비교
- ✓ Feature value는 모두 $[-1,1]$ 로 normalize
- ✓ class의 절반은 target/절반을 outlier로 설정 (바꾸어 한번 더 진행)
- ✓ target을 train/test로 randomly partition 한 뒤 target-test는 outlier와 함께 test set으로 사용(10회 반복)
- ✓ 각 testset 에 대한 f1-score 계산 후 평균값 산출
- ✓ Python 3.7 환경에서 구현

Table 1. Details of benchmark datasets

Dataset	Feature dim.	# of data
Abalone	8	4177
Australian	14	690
Balance	4	625
Glass	9	214
Heart	13	303
Landsat	36	2000
Letter	16	5000
Segment	18	2310
Sonar	60	208
Vehicle	18	846
Waveform3	21	5000
Winequality	11	1599
SVMguide1	4	3089
Diabetes	8	768
Vowel	10	528

Results on benchmark datasets

Table 2. f1-score on benchmark datasets

Dataset	Proposed	SDS [#]	HC [#]	HS [#]	CS [#]	SK [#]	MSML [#]	MIES [#]	QR [#]
Abalone	0.43	0.402	0.511	0.51	0.504	0.497	0.296	0.498	0.512
Australian	0.583	0.592	0.588	0.527	0.574	0.527	0.356	0.576	0.57
Balance	0.726	0.808	0.471	0.471	0.592	0.456	Nan	0.614	0.513
Glass	0.628	0.577	NaN	NaN	0.52	NaN	0.346	0.598	0.634
Heart	0.608	0.567	0.376	0.376	0.568	0.376	0.138	0.565	0.559
Landsat	0.81	0.711	0.665	0.645	0.713	0.695	NaN	0.658	0.636
Letter	0.7	0.601	0.519	0.519	0.515	0.068	0.058	0.514	0.494
Segment	0.775	0.769	0.589	0.588	0.576	0.431	0.436	0.581	0.589
Sonar	0.583	0.506	0.422	0.407	0.506	0.407	0.394	0.505	0.498
Vehicle	0.567	0.651	0.564	0.501	0.497	0.442	NaN	0.505	0.523
Waveform3	0.647	0.674	0.615	0.632	0.639	0.627	NaN	0.637	0.629
Winequality	0.448	0.519	0.5	0.5	0.497	0.226	0.241	0.501	0.501
SVMguide1	0.847	0.838	0.72	0.662	0.727	0.741	0.217	0.755	0.61
Diabetes	0.459	0.508	0.422	0.421	0.299	0.348	NaN	0.499	0.501
Vowel	0.775	0.648	0.34	0.34	0.617	0.338	0.169	0.661	0.648
Average	0.639	0.625	0.521	0.507	0.556	0.441	0.265	0.578	0.561

- ✓ 15개 중 8개의 dataset에 대해 가장 좋은 f1-score 달성
- ✓ 15개 중 9개의 dataset에 대해 SDS보다 나은 f1-score 달성
- ✓ 가장 높은 Average f1-score
- ✓ SDS based method(state-of-the-art)의 성능을 개선 하였음을 확인

Thank you!
