A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks

Kimin Lee, Kibok Lee, Honglak Lee, Jinwoo Shin

Grepp
Zayden

목차

1. Intorduction
2. Mahalanobis distance based classifier
3. Calibration techniques
4. Experiments
5. Q&A

Intorduction

문제 정의

이미지 $P_{\mathbf{X}}$와 $Q_{\mathbf{X}}$가 $P_{\mathbf{X}} Q_{\mathbf{X}} \in \mathcal{X}$일때
혼합 분포를 가지는 새로운 이미지 $\mathbb{P}_{\mathbf{X}|Z} \in \mathcal{X} \times (0, 1)$가 아래와 같을때,

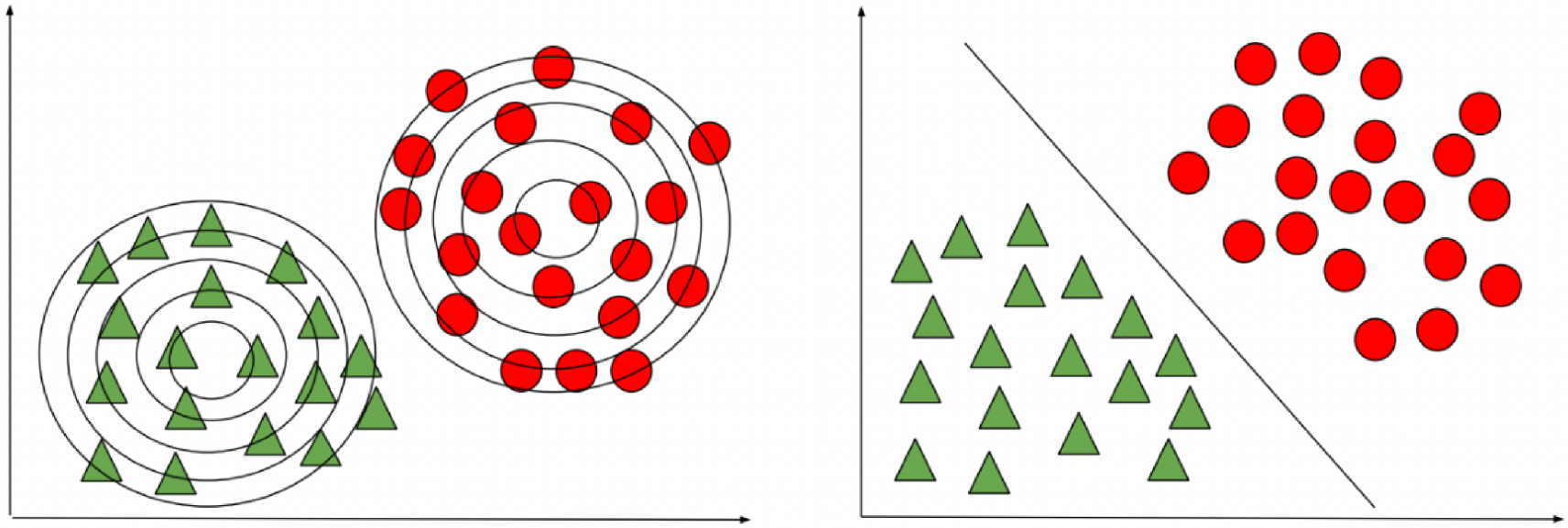
- $\mathbb{P}_{\mathbf{X}|Z=0} = P_{\mathbf{X}}$ → In distribution
- $\mathbb{P}_{\mathbf{X}|Z=1} = Q_{\mathbf{X}}$ → Out distribution

→ $\mathbb{P}_{\mathbf{X}|Z}$ 분포를 가진 이미지 $\mathbf{X}$가 주어지면 이 이미지를 $P_{\mathbf{X}}$인지 아닌지 분류할 수 있을까?

Supervised Learning algorithm

1. **Discriminative Learning Algorithms**
    - 다른 class 사이의 결정 경계를 찾아서 분류를 수행
    - Logistic Regression, Perceptron Algorithm, etc ...
2. **Generative Learning Algorithms**
    - 각 class의 분포(가우시안 분포를 가정)를 이용해서 class를 분류
    - GDA

GLA vs DLA



사후 확률 (Posterior distribution)

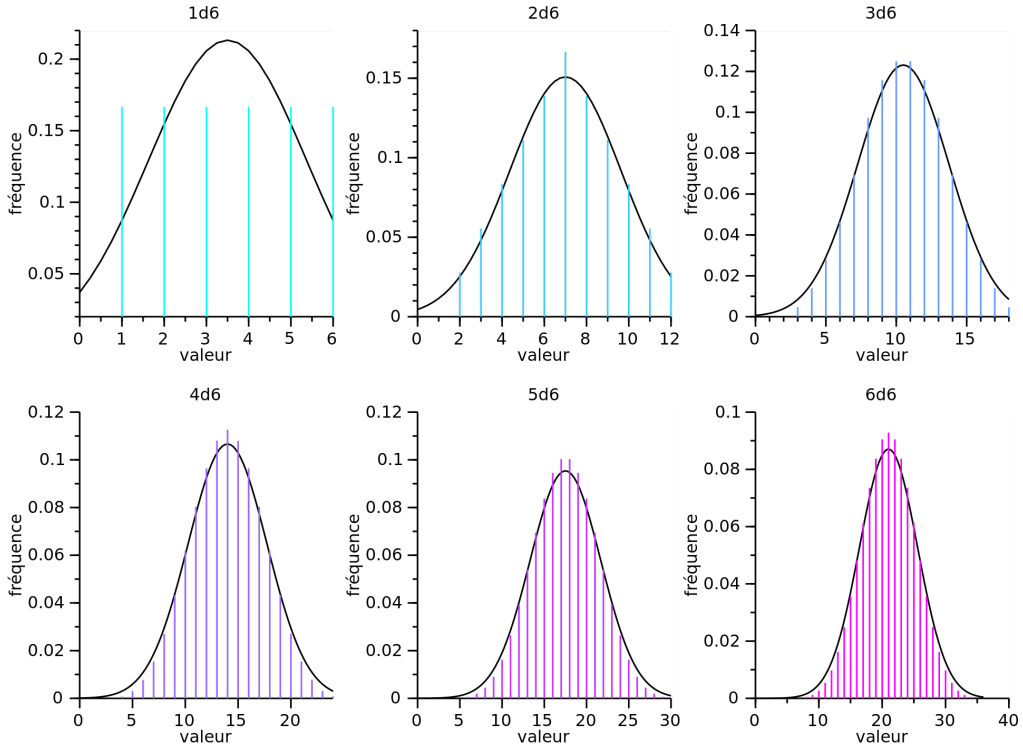
softmax 기반 분류 모델의 사후 확률은 softmax classifier로 이루어져있음.

$$P(y=c|x)=\frac{\exp(\mathrm{w}_c^{\mathsf{T}} f(\mathrm{x})+b_c)}{\sum_{c'}\exp(\mathrm{w}_{c'}^{\mathsf{T}} f(\mathrm{x})+b_{c'})},$$

→ 이를 Malhalanobis distance 기반 classifier로 대체하여 confidence score $M(\mathrm{x})$ 와 추측 class $\hat{y}(\mathrm{x})$를 구한다.

중심 극한 정리 (린데베르그-레비)

확률변수 $X_1, X_2, \ldots$ 들이 서로 독립적이고, 같은 확률 분포를 가지고 그 확률분포의 기대값(μ)과 표준편차(σ)가 유한하면 $X_1, X_2, \ldots$ 평균의 분포는 정규분포를 따른다.



GDA에서의 사후 확률

- 사후 확률은 사전 확률과 가능도의 곱으로 나타낼 수 있음.
  $P(w_i|x) = P(w_i)\, p(x|w_i)$

- w_i가 N차원의 공간에서 가우시안 분포를 가질때 가능도 $P(w_i)$ (likelihood)

$$p(x|w_i) = N(\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i))$$

$$P(w_i|x) = \ln(f(x)) = \ln(p(x|w_i)P(w_i)) = \ln(N(\mu_i, \Sigma_i)) + \ln(P(w_i))$$

클래스 별 prior와 covariance가 동일하다면 상수항 제거

$$P(w_i|x) = -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - \frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln(|\Sigma_i|) + \ln(P(w_i))$$

Mahalanobis Distance

x가 주어졌을때 평균 벡터 μ_i와 공분산 행렬 Σ_i를 갖는 가우시안 분포와의 거리

$$((x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i))^{0.5}$$

Mahalanobis distance based classifier

학습된 모델에 Mahalanobis Distance를 어떻게 적용시킬까?

- softmax classifier로 학습된 모델에 사용된 training data 샘플로부터 각 class의 평균과 전체 데이터에 대한 공분산 행렬을 계산

$$\hat{\mu}_c = \frac{1}{N_c} \sum_{i:y_i=c} f(\mathbf{x}_i),$$

$$\hat{\Sigma} = \sum_{c} \sum_{i:y_i=c} (f(\mathbf{x}_i) - \hat{\mu}_c)(f(\mathbf{x}_i) - \hat{\mu}_c)^\mathsf{T}$$

Mahalanobis distance-based confidence score

confidence score $M(\mathbf{x})$는 가장 가까운 class 분포와 Mahalanobis distance 로 계산

$$M(\mathbf{x}) = \max_c -(f(\mathbf{x}_i) - \hat{\mu}_c)^\intercal \hat{\Sigma}^{-1}(f(\mathbf{x}_i) - \hat{\mu}_c).$$

→ 거리가 가까워지면 (작아지면) confidences score는 더 커짐

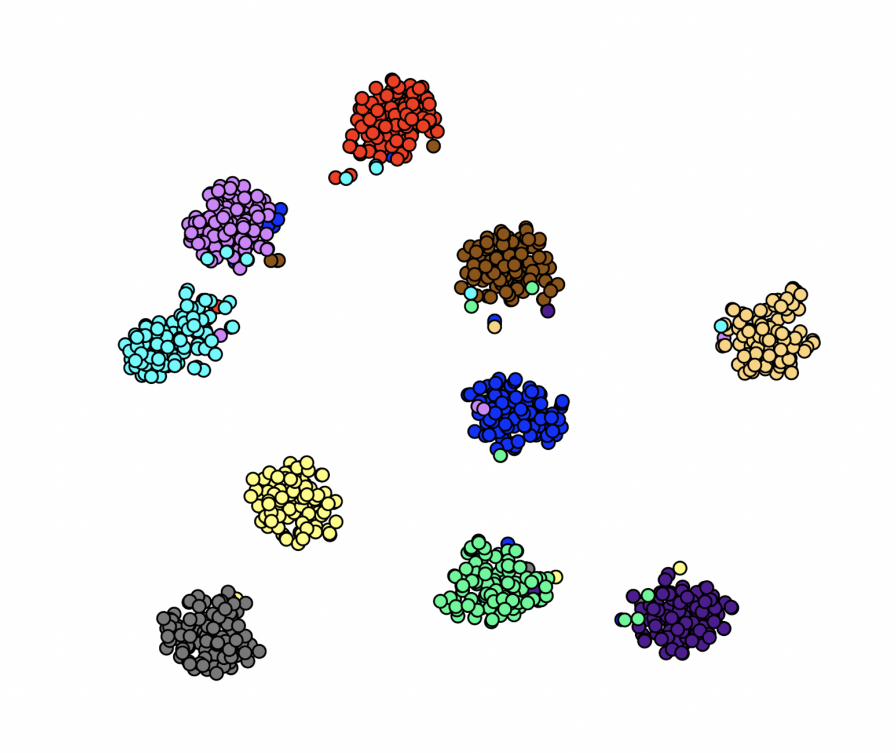
Mahalanobis distance based classifier

추측 class $\hat{y}(\mathrm{x})$는 거리가 가장 가까운 class conditional distribution의 class를 할당

$$\hat{y}(\mathrm{x}) = \arg\min_{c} (f(\mathrm{x}) - \hat{\mu}_c)^\intercal \hat{\Sigma}^{-1} (f(\mathrm{x}) - \hat{\mu}_c).$$

OOD detection

이때 $M(\mathbf{x})$가 특정 threshold 보다 높은 경우 $\mathbf{x}$는 positive로 결정(i.e. in-distribution) 낮은 경우는 OoD로 결정



Calibration techniques

Input pre-processing

입력 이미지에 대한 softmax score를 높이는 방향으로 perturbation을 추가한다.

- **ODIN**

$$\tilde{x} = x - \epsilon sign(-\nabla_x log S_{\hat{y}}(x;T)),$$

- **Here**

$$\begin{aligned}\hat{\mathbf{x}} &= \mathbf{x} - \epsilon sign(-\nabla_{\mathbf{x}} M(\mathbf{x})) \\ &= \mathbf{x} - \epsilon sign(\nabla_{\mathbf{x}}(f(\mathbf{x}_i) - \hat{\mu}_{\hat{c}})^\intercal \hat{\Sigma}^{-1}(f(\mathbf{x}_i) - \hat{\mu}_{\hat{c}})),\end{aligned}$$

Feature ensemble

low-level feature를 추가적으로 뽑아내서 성능을 더 향상시킴

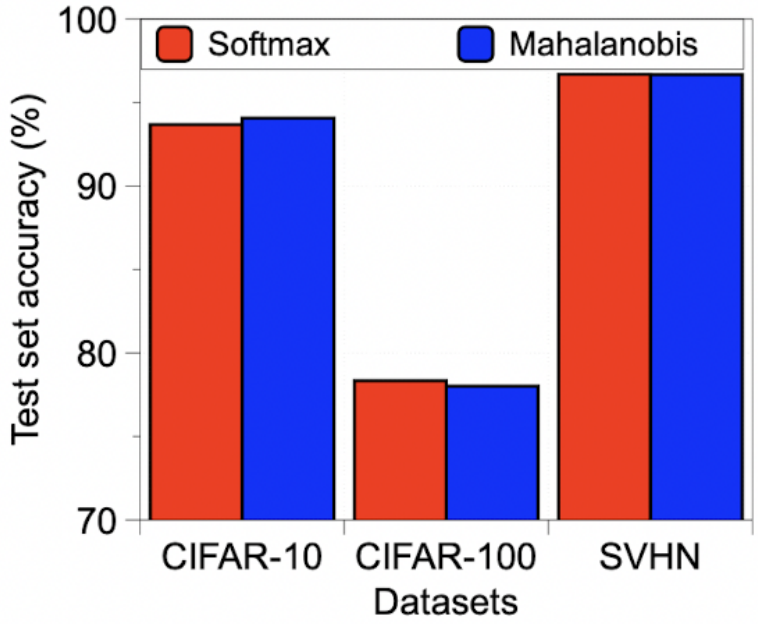
ℓ번째 히든 feature를 $f_\ell(\mathbf{x})$라고 하면, 클래스 평균$\hat{\mu_{\ell,c}}$과 tied(동일한 shape을 가진) 공분산$\hat{\Sigma}_\ell$을 계산하고 각 테스트 샘플 $\mathbf{x}$에 대해 ℓ번째 레이어의 confidence score를 계산.

Algorithm 1 Computing the Mahalanobis distance-based confidence score.

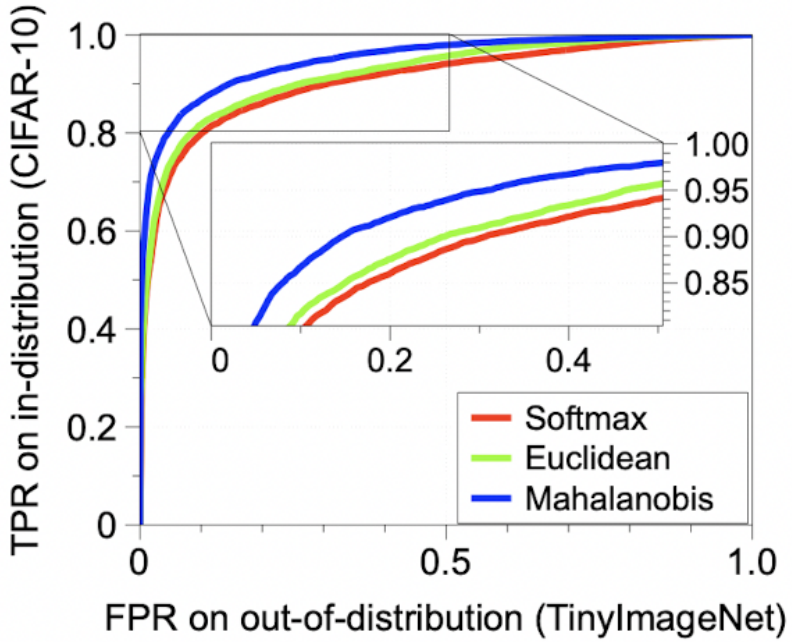
Input: Test sample $\mathbf{x}$, weights of logistic regression detector α_ℓ, noise ε and parameters of Gaussian distributions $\{\widehat{\mu}_{\ell,c}, \widehat{\mathbf{\Sigma}}_\ell : \forall \ell, c\}$

Initialize score vectors: $\mathbf{M}(\mathbf{x}) = [M_\ell : \forall \ell]$
for each layer $\ell \in 1, \ldots, L$ **do**
  Find the closest class: $\widehat{c} = \arg\min_c \ (f_\ell(\mathbf{x}) - \widehat{\mu}_{\ell,c})^\top \widehat{\mathbf{\Sigma}}_\ell^{-1} (f_\ell(\mathbf{x}) - \widehat{\mu}_{\ell,c})$
  Add small noise to test sample: $\widehat{\mathbf{x}} = \mathbf{x} - \varepsilon \text{sign}\left(\nabla_{\mathbf{x}} \left(f_\ell(\mathbf{x}) - \widehat{\mu}_{\ell,\widehat{c}}\right)^\top \widehat{\mathbf{\Sigma}}_\ell^{-1} \left(f_\ell(\mathbf{x}) - \widehat{\mu}_{\ell,\widehat{c}}\right)\right)$
  Computing confidence score: $M_\ell = \max_c - \left(f_\ell(\widehat{\mathbf{x}}) - \widehat{\mu}_{\ell,c}\right)^\top \widehat{\mathbf{\Sigma}}_\ell^{-1} \left(f_\ell(\widehat{\mathbf{x}}) - \widehat{\mu}_{\ell,c}\right)$
end for
return Confidence score for test sample $\sum_\ell \alpha_\ell M_\ell$

- 마지막 레이어에 비해서 더 좋은 교정된 결과를 얻는 경우가 있다.
- 논문에서 진행한 실험에서는 모든 dense 또는 residual block의 마지막에서 confidence score를 계산하였고 average pooling을 통해서 $\mathcal{F} \times \mathcal{H} \times \mathcal{W} \rightarrow \mathcal{F} \times 1$으로 크기를 줄이고 logistic regression을 통해서 confidence score를 계산한다.
    - logistic regression detector는 in-distribution 및 OoD로 이루어진 valdiation set을 교차 검증해서 학습된다.



(b) Classification accuracy



(c) ROC curve

Experiments

Method	Feature ensemble	Input pre-processing	TNR at TPR 95%	AUROC	Detection accuracy	AUPR in	AUPR out
Baseline [13]	-	-	32.47	89.88	85.06	85.40	93.96
ODIN [21]	-	-	86.55	96.65	91.08	92.54	98.52
Mahalanobis (ours)	-	-	54.51	93.92	89.13	91.56	95.95
	-	✓	92.26	98.30	93.72	96.01	99.28
	✓	-	91.45	98.37	93.55	96.43	99.35
	✓	✓	**96.42**	**99.14**	**95.75**	**98.26**	**99.60**

In-dist (model)	OOD	Validation on OOD samples			Validation on adversarial samples		
		TNR at TPR 95%	AUROC	Detection acc.	TNR at TPR 95%	AUROC	Detection acc.
		Baseline [13] / ODIN [21] / Mahalanobis (ours)			Baseline [13] / ODIN [21] / Mahalanobis (ours)		
CIFAR-10 (DenseNet)	SVHN	40.2 / 86.2 / **90.8**	89.9 / 95.5 / **98.1**	83.2 / 91.4 / **93.9**	40.2 / 70.5 / **89.6**	89.9 / 92.8 / **97.6**	83.2 / 86.5 / **92.6**
	TinyImageNet	58.9 / 92.4 / **95.0**	94.1 / 98.5 / **98.8**	88.5 / 93.9 / **95.0**	58.9 / 87.1 / **94.9**	94.1 / 97.2 / **98.8**	88.5 / 92.1 / **95.0**
	LSUN	66.6 / 96.2 / **97.2**	95.4 / 99.2 / **99.3**	90.3 / 95.7 / **96.3**	66.6 / 92.9 / **97.2**	95.4 / 98.5 / **99.2**	90.3 / 94.3 / **96.2**
CIFAR-100 (DenseNet)	SVHN	26.7 / 70.6 / **82.5**	82.7 / 93.8 / **97.2**	75.6 / 86.6 / **91.5**	26.7 / 39.8 / **62.2**	82.7 / 88.2 / **91.8**	75.6 / 80.7 / **84.6**
	TinyImageNet	17.6 / 42.6 / **86.6**	71.7 / 85.2 / **97.4**	65.7 / 77.0 / **92.2**	17.6 / 43.2 / **87.2**	71.7 / 85.3 / **97.0**	65.7 / 77.2 / **91.8**
	LSUN	16.7 / 41.2 / **91.4**	70.8 / 85.5 / **98.0**	64.9 / 77.1 / **93.9**	16.7 / 42.1 / **91.4**	70.8 / 85.7 / **97.9**	64.9 / 77.3 / **93.8**
SVHN (DenseNet)	CIFAR-10	69.3 / 71.7 / **96.8**	91.9 / 91.4 / **98.9**	86.6 / 85.8 / **95.9**	69.3 / 69.3 / **97.5**	91.9 / 91.9 / **98.8**	86.6 / 86.6 / **96.3**
	TinyImageNet	79.8 / 84.1 / **99.9**	94.8 / 95.1 / **99.9**	90.2 / 90.4 / **98.9**	79.8 / 79.8 / **99.9**	94.8 / 94.8 / **99.8**	90.2 / 90.2 / **98.9**
	LSUN	77.1 / 81.1 / **100**	94.1 / 94.5 / **99.9**	89.1 / 89.2 / **99.3**	77.1 / 77.1 / **100**	94.1 / 94.1 / **99.9**	89.1 / 89.1 / **99.2**
CIFAR-10 (ResNet)	SVHN	32.5 / 86.6 / **96.4**	89.9 / 96.7 / **99.1**	85.1 / 91.1 / **95.8**	32.5 / 40.3 / **75.8**	89.9 / 86.5 / **95.5**	85.1 / 77.8 / **89.1**
	TinyImageNet	44.7 / 72.5 / **97.1**	91.0 / 94.0 / **99.5**	85.1 / 86.5 / **96.3**	44.7 / 69.6 / **95.5**	91.0 / 93.9 / **99.0**	85.1 / 86.0 / **95.4**
	LSUN	45.4 / 73.8 / **98.9**	91.0 / 94.1 / **99.7**	85.3 / 86.7 / **97.7**	45.4 / 70.0 / **98.1**	91.0 / 93.7 / **99.5**	85.3 / 85.8 / **97.2**
CIFAR-100 (ResNet)	SVHN	20.3 / 62.7 / **91.9**	79.5 / 93.9 / **98.4**	73.2 / 88.0 / **93.7**	20.3 / 12.2 / **41.9**	79.5 / 72.0 / **84.4**	73.2 / 67.7 / **76.5**
	TinyImageNet	20.4 / 49.2 / **90.9**	77.2 / 87.6 / **98.2**	70.8 / 80.1 / **93.3**	20.4 / 33.5 / **70.3**	77.2 / 83.6 / **87.9**	70.8 / 75.9 / **84.6**
	LSUN	18.8 / 45.6 / **90.9**	75.8 / 85.6 / **98.2**	69.9 / 78.3 / **93.5**	18.8 / 31.6 / **56.6**	75.8 / 81.9 / **82.3**	69.9 / 74.6 / **79.7**
SVHN (ResNet)	CIFAR-10	78.3 / 79.8 / **98.4**	92.9 / 92.1 / **99.3**	90.0 / 89.4 / **96.9**	78.3 / 79.8 / **94.1**	92.9 / 92.1 / **97.6**	90.0 / 89.4 / **94.6**
	TinyImageNet	79.0 / 82.1 / **99.9**	93.5 / 92.0 / **99.9**	90.4 / 89.4 / **99.1**	79.0 / 80.5 / **99.2**	93.5 / 92.9 / **99.3**	90.4 / 90.1 / **98.8**
	LSUN	74.3 / 77.3 / **99.9**	91.6 / 89.4 / **99.9**	89.0 / 87.2 / **99.5**	74.3 / 76.3 / **99.9**	91.6 / 90.7 / **99.9**	89.0 / 88.2 / **99.5**

class incremental learning task

새로운 class가 추가되었을때 학습시키는 방법

Algorithm 2 Updating Mahalanobis distance-based classifier for class-incremental learning.

Input: set of samples from a new class $\{\mathbf{x}_i : \forall i = 1 \dots N_{C+1}\}$, mean and covariance of observed classes $\{\widehat{\mu}_c : \forall c = 1 \dots C\}, \widehat{\mathbf{\Sigma}}$

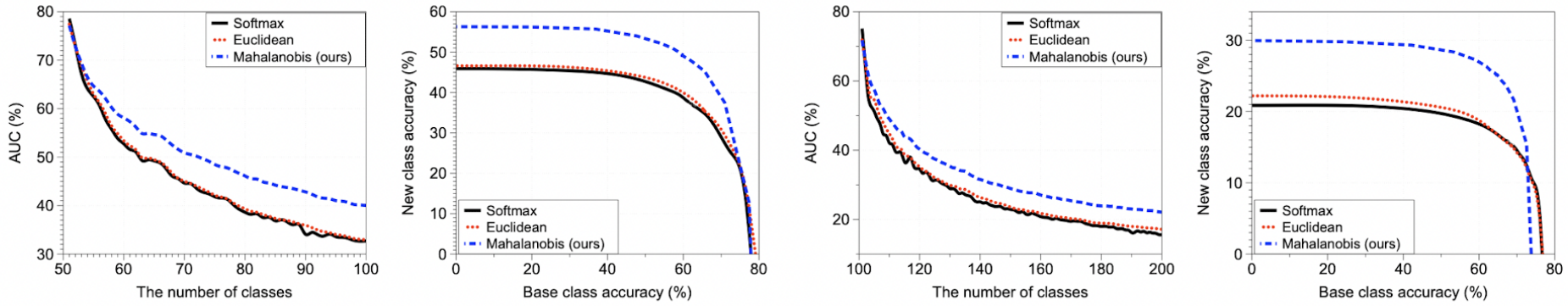
Compute the new class mean: $\widehat{\mu}_{C+1} \leftarrow \frac{1}{N_{C+1}} \sum_i f(\mathbf{x}_i)$

Compute the covariance of the new class: $\widehat{\mathbf{\Sigma}}_{C+1} \leftarrow \frac{1}{N_{C+1}} \sum_i (f(\mathbf{x}_i) - \widehat{\mu}_{C+1})(f(\mathbf{x}_i) - \widehat{\mu}_{C+1})^\top$

Update the shared covariance: $\widehat{\mathbf{\Sigma}} \leftarrow \frac{C}{C+1}\widehat{\mathbf{\Sigma}} + \frac{1}{C+1}\widehat{\mathbf{\Sigma}}_{C+1}$

return Mean and covariance of all classes $\{\widehat{\mu}_c : \forall c = 1 \dots C+1\}, \widehat{\mathbf{\Sigma}}$

- Softmax는 [4]의 방법으로, Euclidean은 class 평균만 가지고 계산



(a) Base: half of CIFAR-100 / New: the other half

(b) Base: CIFAR-100 / New: ImageNet

Figure 4: Experimental results of class-incremental learning on CIFAR-100 and ImageNet datasets. In each experiment, we report (left) AUC with respect to the number of learned classes and, (right) the base-new class accuracy curve after the last new classes is added.

Q&A

References

- [1] www.geeksforgeeks.org/gaussian-discriminant-analysis/
- [2] gaussian37.github.io/ml-concept-gaussian_discriminant/
- [3] github.com/pokaxpoka/deep_Mahalanobis_detector/blob/90c2105e78c6f76a2801fc4c1cb1b84f4ff9af63/lib_generation.py
- [4] Mensink, Thomas, Verbeek, Jakob, Perronnin, Florent, and Csurka, Gabriela. Distance-based image classification: Generalizing to new classes at near-zero cost. IEEE transactions on pattern analysis and machine intelligence, 2013.