

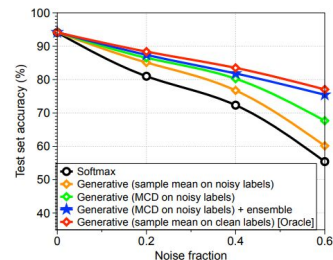
Robust Inference via Generative Classifiers for Handling Noisy Labels

Lee., KAIST / ICML 2019 (Long oral)

Presenter: Kangyeol Kim, Korea University

Motivation and questions

- For handling noisy datasets, this paper adapts generative classifiers framework.
- First, what is noisy dataset?
 - The quality of image or text (such as comments on web) is bad.
 - There exists incorrectly labeled data.
- Second, is it really problem? Absolutely. see next graph.
- How can we tackle it then ?
 - The concept called 'robust' is leveraged for this
 - Robust + (training, model, inference...)



(a) Test set accuracy comparison

Measure of robustness

- Breakdown point (source: wikipedia)

Breakdown point [\[edit\]](#)

Intuitively, the breakdown point of an [estimator](#) is the proportion of incorrect observations (e.g. arbitrarily large observations) an estimator can handle before giving an incorrect (e.g., arbitrarily large) result. For example, given n independent random variables (X_1, \dots, X_n) and the corresponding realizations x_1, \dots, x_n , we can use $\bar{X}_n := \frac{X_1 + \dots + X_n}{n}$ to estimate the mean. Such an estimator has a breakdown point of 0 because we can make \bar{x} arbitrarily large just by changing any of x_1, \dots, x_n .

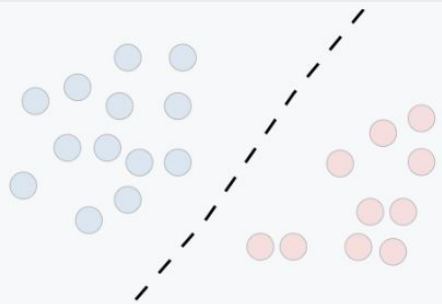
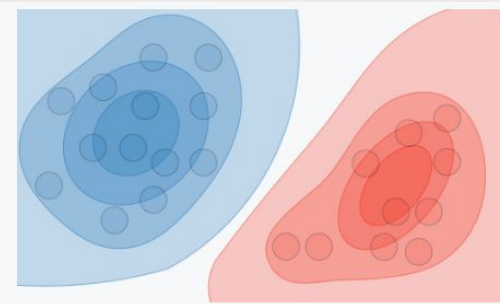
The higher the breakdown point of an estimator, the more robust it is. Intuitively, we can understand that a breakdown point cannot exceed 50% because if more than half of the observations are contaminated, it is not possible to distinguish between the underlying distribution and the contaminating distribution [Rousseeuw & Leroy \(1986\)](#). Therefore, the maximum breakdown point is 0.5 and there are estimators which achieve such a breakdown point. For example, the median has a breakdown point of 0.5. The $X\%$ trimmed mean has breakdown point of $X\%$, for the chosen level of X . [Huber \(1981\)](#) and [Maronna, Martin & Yohai \(2006\)](#) contain more details. The level and the power breakdown points of tests are investigated in [He, Simpson & Portnoy \(1990\)](#).

Statistics with high breakdown points are sometimes called **resistant statistics**.^[4]

Discriminative vs Generative model

- For utilizing 'Breakdown point', this paper simply **interprets** the general classifier with softmax layer into generative model form.

●

| | Discriminative model | Generative model |
|----------------|---|--|
| Goal | Directly estimate $P(y x)$ | Estimate $P(x y)$ to then deduce $P(y x)$ |
| What's learned | Decision boundary | Probability distributions of the data |
| Illustration |  |  |
| Examples | Regressions, SVMs | GDA, Naive Bayes |

Discriminative vs Generative model

- In short, in generative modeling, we learn $p(x|y), p(y)$ and distill $p(y|x)$.
- Specifically, the paper utilizes linear discriminant analysis (LDA), by assuming the features from penultimate layer follows class-conditional gaussian distribution which has tied covariance 1). And assume a Bernoulli distribution of the class prior: $P(f(x)|y = c) = N(f(x)|\mu_c, \Sigma), P(y = c) = \beta_c$
- Then, we easily have:

$$\begin{aligned} P(y = c | f(\mathbf{x})) &= \frac{P(y = c) P(f(\mathbf{x}) | y = c)}{\sum_{c'} P(y = c') P(f(\mathbf{x}) | y = c')} \\ &= \frac{\exp(\mu_c^\top \Sigma^{-1} f(\mathbf{x}) - \frac{1}{2} \mu_c^\top \Sigma^{-1} \mu_c + \log \beta_c)}{\sum_{c'} \exp(\mu_{c'}^\top \Sigma^{-1} f(\mathbf{x}) - \frac{1}{2} \mu_{c'}^\top \Sigma^{-1} \mu_{c'} + \log \beta_{c'})}. \end{aligned}$$

1) Analytic justification is shown in the paper

Discriminative vs Generative model

- In short, in generative modeling, we learn $p(x|y), p(y)$ and distill $p(y|x)$.
- Specifically, the paper utilizes linear discriminant analysis (LDA), by assuming the features from penultimate layer follows class-conditional gaussian distribution which has tied covariance 1). And assume a Bernoulli distribution of the class prior: $P(f(x)|y = c) = N(f(x)|\mu_c, \Sigma), P(y = c) = \beta_c$
- Then, we easily have:

$$\begin{aligned} P(y = c | f(\mathbf{x})) &= \frac{P(y = c) P(f(\mathbf{x}) | y = c)}{\sum_{c'} P(y = c') P(f(\mathbf{x}) | y = c')} \\ &= \frac{\exp(\mu_c^\top \Sigma^{-1} f(\mathbf{x}) - \frac{1}{2} \mu_c^\top \Sigma^{-1} \mu_c + \log \beta_c)}{\sum_{c'} \exp(\mu_{c'}^\top \Sigma^{-1} f(\mathbf{x}) - \frac{1}{2} \mu_{c'}^\top \Sigma^{-1} \mu_{c'} + \log \beta_{c'})}. \end{aligned}$$

1) Analytic justification is shown in the paper

Bridge between softmax classifier / LDA

$$P(y = c | \mathbf{x}) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x} + b_c)}{\sum_{c'} \exp(\mathbf{w}_{c'}^\top \mathbf{x} + b_{c'})}$$

$$P(y = c | \mathbf{x}) = \frac{P(y = c) P(\mathbf{x} | y = c)}{\sum_{c'} P(y = c') P(\mathbf{x} | y = c')} = \frac{\exp(\mu_c^\top \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_c^\top \Sigma^{-1} \mu_c + \log \beta_c)}{\sum_{c'} \exp(\mu_{c'}^\top \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_{c'}^\top \Sigma^{-1} \mu_{c'} + \log \beta_{c'})}.$$

- Softmax classifier can be interpreted as finding posterior distribution of generative classifier.
- This implies \mathbf{x} *might be* fitted in Gaussian distribution:

$$P(x | y = c) = N(x | \mu_c, \Sigma), P(y = c) = \beta_c$$

Estimating parameters of GM

$$P(y = c | \mathbf{x}) = \frac{P(y = c) P(\mathbf{x} | y = c)}{\sum_{c'} P(y = c') P(\mathbf{x} | y = c')} = \frac{\exp(\mu_c^\top \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_c^\top \Sigma^{-1} \mu_c + \log \beta_c)}{\sum_{c'} \exp(\mu_{c'}^\top \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_{c'}^\top \Sigma^{-1} \mu_{c'} + \log \beta_{c'})}.$$

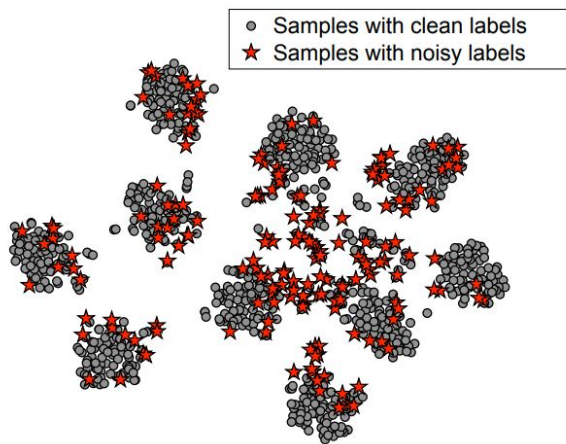
- From above equation, we should estimate *mean* and *covariance* at each class. By simply putting it as estimating the sample class *mean* and *covariance*, we have:

$$\bar{\mu}_c = \sum_{i: y_i = c} \frac{f(\mathbf{x}_i)}{N_c}, \quad \bar{\beta}_c = \frac{N_c}{N},$$
$$\bar{\Sigma} = \sum_c \sum_{i: y_i = c} \frac{(f(\mathbf{x}_i) - \bar{\mu}_c)(f(\mathbf{x}_i) - \bar{\mu}_c)^\top}{N}$$

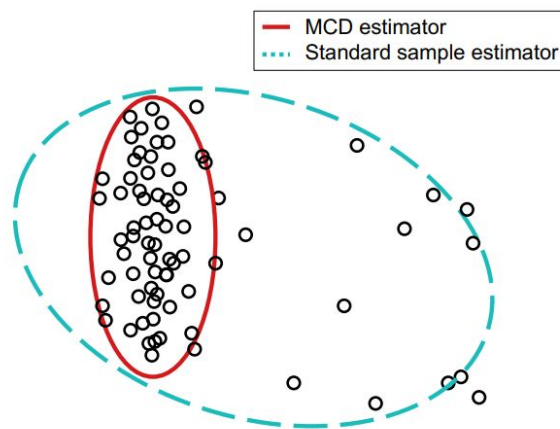
- However, the above estimators are prone to be highly influenced by outliers. We already know that *mean* is vulnerable to outliers. And we can measure it via **breakdown point**.
- Now, how can we obtain estimators which have high **breakdown point**

Alternative to estimate parameters of GM

- For this, **minimum covariance determinant (MCD)** estimator is used instead.
- Here, the assumption is that corrupted dataset can be trained as **(b)** and the objective of MCD estimator can be seen as **(c)** in high view.



(b) Penultimate features by t-SNE



(c) An illustration of the MCD estimator

Alternative to estimate parameters of GM

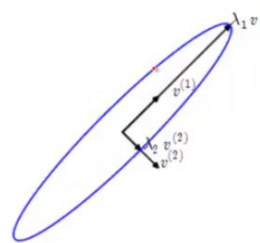
- In the viewpoint of objective function, MCD estimator aims at minimizing the sample covariance.

$$\min_{\mathcal{X}_{K_c} \subset \mathcal{X}_{N_c}} \det(\widehat{\Sigma}_c) \quad \text{subject to } |\mathcal{X}_{K_c}| = K_c,$$

- In my intuition, since *determinant* is a product of eigenvalues, it is similar to reducing an exaggerated scale which caused by outliers
- In fact, MCD estimator has high breakdown points. (see paper for proof)
- The MCD estimator for the generative classifier under LDA assumption is known to attain near optimal breakdown value of

$$\min_c \frac{\lfloor (N_c - d + 1)/2 \rfloor}{N_c} \approx 50\%$$

- Thus, the paper chooses $\lfloor (N_c + d + 1)/2 \rfloor$ as K_c



Approximation Algorithm for MCD

Algorithm 1 (Rousseeuw & Driessen, 1999) Approximating MCD for a single Gaussian.

- 1: **Input:** $\mathcal{X}_{N_c} = \{\mathbf{x}_i : i = 1, \dots, N_c\}$ and the maximum number of iterations I_{\max} .
- 2: Uniformly sample initial subset $\mathcal{X}_{K_c} \subset \mathcal{X}_{N_c}$, where $|\mathcal{X}_{K_c}| = \lfloor (N_c + d + 1)/2 \rfloor$.
- 3: Compute a mean $\hat{\mu}_c = \frac{1}{|\mathcal{X}_{K_c}|} \sum_{\mathbf{x} \in \mathcal{X}_{K_c}} f(\mathbf{x})$, and covariance $\hat{\Sigma}_c = \frac{1}{|\mathcal{X}_{K_c}|} \sum_{\mathbf{x} \in \mathcal{X}_{K_c}} (f(\mathbf{x}) - \hat{\mu}_c)(f(\mathbf{x}) - \hat{\mu}_c)^\top$.
- 4: **for** $i = 1$ **to** I_{\max} **do**
- 5: Compute the Mahalanobis distance for all $\mathbf{x} \in \mathcal{X}_{N_c}$:
 $\alpha(\mathbf{x}) = (f(\mathbf{x}) - \hat{\mu}_c)^\top \hat{\Sigma}_c^{-1} (f(\mathbf{x}) - \hat{\mu}_c)$.
- 6: Update \mathcal{X}_{K_c} such that it includes $\lfloor (N_c + d + 1)/2 \rfloor$ samples with smallest distance $\alpha(\mathbf{x})$.
- 7: Compute sample mean and covariance, i.e., $\hat{\mu}_c, \hat{\Sigma}_c$, using new subset \mathcal{X}_{K_c} .
- 8: Exit the loop if the determinant of covariance matrix is not decreasing anymore.
- 9: **end for**
- 10: Return $\hat{\mu}_c$ and $\hat{\Sigma}_c$

Step - 1



$$\hat{\Sigma} = \frac{\sum_c K_c \hat{\Sigma}_c}{\sum_c K_c}$$

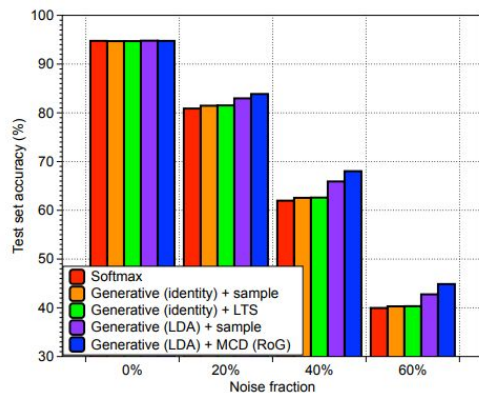
Step - 2

Ensemble of Generative Classifiers

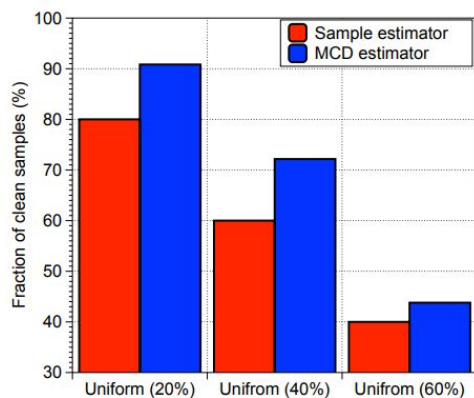
3.3. Ensemble of Generative Classifiers

To further improve the performance of our method, we consider the ensemble of generative classifiers not only from the penultimate features but also from other low-level features in DNNs. Formally, given training data, we extract ℓ -th hidden features of DNNs, denoted by $f_\ell(\mathbf{x}) \in \mathbb{R}^{d_\ell}$, and compute the corresponding parameters of a generative classifier (i.e., $\hat{\mu}_{\ell,c}$ and $\hat{\Sigma}_\ell$) using the (approximated version of) MCD estimator. Then, the final posterior distribution is obtained by the weighted sum of all posterior distributions of generative classifiers: $\sum_{\ell} \alpha_\ell P(y = c | f_\ell(\mathbf{x}))$, where α_ℓ is an ensemble weight at ℓ -th layer. In our experiments, we choose the weight of each layer by optimizing negative

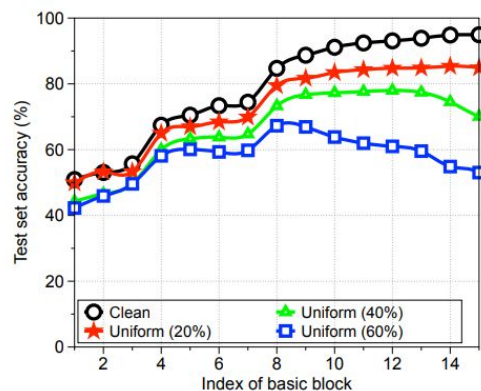
Ensemble of Generative Classifiers



(a) Model comparison



(b) Accuracy of the MCD estimator



(c) Layer-wise accuracy

Experiment of ensemble effect

| Model | Inference method | Ensemble | Clean | Uniform (20%) | Uniform (40%) | Uniform (60%) |
|----------|-------------------------|----------|-------|---------------|---------------|---------------|
| DenseNet | Softmax | - | 94.11 | 81.01 | 72.34 | 55.42 |
| | Generative + sample | - | 94.18 | 85.12 | 76.75 | 60.14 |
| | | ✓ | 93.97 | 87.40 | 81.27 | 69.81 |
| | Generative + MCD (ours) | - | 94.22 | 86.54 | 80.27 | 67.67 |
| | | ✓ | 94.18 | 87.41 | 81.83 | 75.45 |
| ResNet | Softmax | - | 94.76 | 80.88 | 61.98 | 39.96 |
| | Generative + sample | - | 94.80 | 82.97 | 65.92 | 42.76 |
| | | ✓ | 94.82 | 83.36 | 68.57 | 46.45 |
| | Generative + MCD (ours) | - | 94.76 | 83.86 | 68.03 | 44.87 |
| | | ✓ | 94.68 | 84.62 | 75.28 | 54.57 |

Table 1. Effects of an ensemble method. We use the CIFAR-10 dataset with various uniform noise fractions. All values are percentages and the best results are highlighted in bold if the gain is bigger than 1% compared to softmax classifier.

Experiment on various datasets

| Noise type (%) | ResNet | | | DenseNet | | |
|----------------|----------------------|----------------------------|----------------------|----------------------|----------------------------|----------------------|
| | CIFAR-10 | CIFAR-100 Softmax / RoG | SVHN | CIFAR-10 | CIFAR-100 Softmax / RoG | SVHN |
| Clean | 94.76 / 94.68 | 76.81 / 76.97 | 95.96 / 96.09 | 94.11 / 94.18 | 75.69 / 72.67 | 96.59 / 96.18 |
| Uniform (20%) | 80.88 / 84.62 | 64.43 / 68.29 | 83.52 / 91.67 | 81.01 / 87.41 | 61.72 / 64.29 | 86.92 / 89.50 |
| Uniform (40%) | 61.98 / 75.28 | 48.62 / 60.76 | 72.89 / 87.16 | 72.34 / 81.83 | 50.89 / 55.68 | 81.91 / 85.71 |
| Uniform (60%) | 39.96 / 54.57 | 27.57 / 48.42 | 61.23 / 80.52 | 55.42 / 75.45 | 38.33 / 44.12 | 71.18 / 77.67 |
| Flip (20%) | 79.65 / 88.73 | 65.14 / 73.37 | 85.49 / 93.00 | 79.18 / 91.23 | 65.37 / 69.03 | 95.04 / 94.86 |
| Flip (40%) | 58.13 / 61.56 | 46.61 / 66.71 | 65.88 / 87.96 | 56.29 / 86.42 | 46.04 / 69.38 | 88.83 / 93.57 |

Experiment on NLP datasets

- Noise is added to latent vectors here.

| Dataset | Training method | Softmax / RoG | | | |
|---------|-----------------|----------------------|----------------------|----------------------|----------------------|
| | | Clean | Uniform (20%) | Uniform (40%) | Uniform (60%) |
| Twitter | Cross-entropy | 87.47 / 85.28 | 79.13 / 81.66 | 66.74 / 79.37 | 50.83 / 73.65 |
| | Forward (gold) | 78.07 / 83.59 | 72.97 / 81.60 | 64.55 / 78.24 | 51.59 / 72.33 |
| | GLC | 83.47 / 84.68 | 66.09 / 81.66 | 59.72 / 79.00 | 53.14 / 72.93 |
| Reuters | Cross-entropy | 95.88 / 94.77 | 87.74 / 92.83 | 76.54 / 82.20 | 57.49 / 64.98 |
| | Forward (gold) | 94.57 / 94.75 | 88.44 / 93.24 | 77.85 / 82.56 | 61.01 / 66.56 |
| | GLC | 95.97 / 94.91 | 81.45 / 92.75 | 73.40 / 83.82 | 59.21 / 67.91 |