Sixth Conference on Discrete Optimization and Machine Learning, Tokyo July 8, 2024

Training Error and Bayes Error in Deep Learning

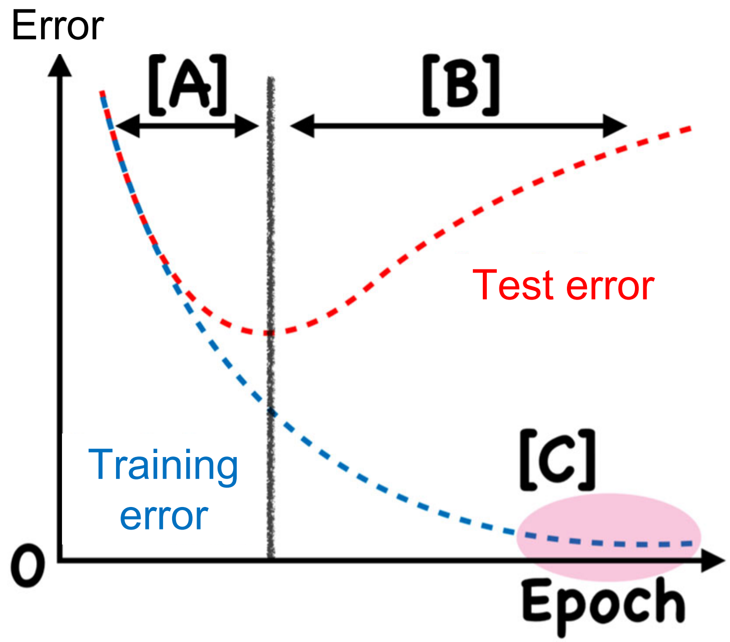
Masashi Sugiyama

RIKEN Center for Advanced Intelligence Project/
The University of Tokyo, Japan

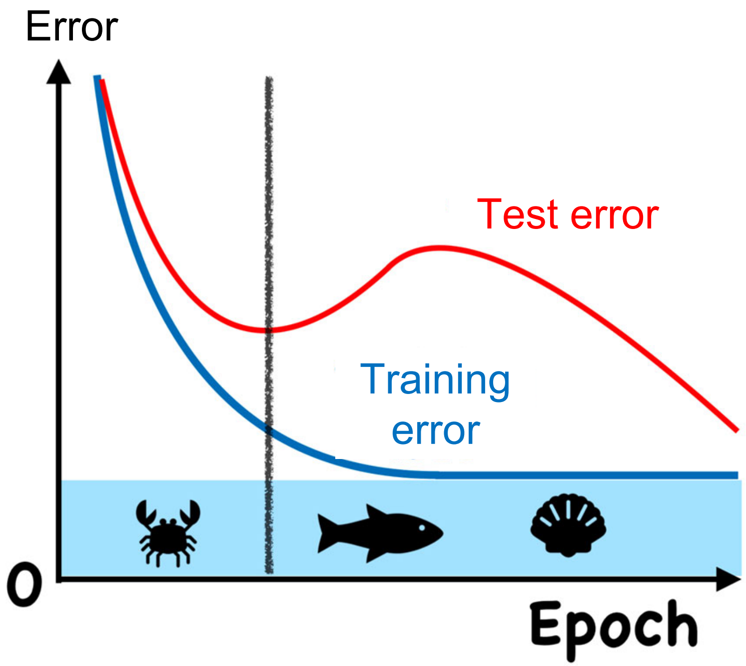
http://www.ms.k.u-tokyo.ac.jp/sugi/

This Talk in a Nutshell

- Overfitting: Too small training error can yield a large test error.
  - With deep learning, it is easy to achieve zero training error.
  - But the minimum achievable test error is not necessarily zero.



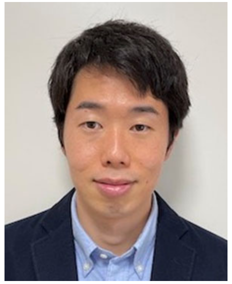
This Talk in a Nutshell



- In this talk, we discuss:
  1. Can we mitigate overfitting by avoiding too small training error?
  2. Can we estimate the Bayes error accurately?

- References:
  - Ishida, T., Yamane, I., Sakai, T., Niu, G. & Sugiyama, M. (ICML2020).
  - Ishida, T., Yamane, I., Charoenphakdee, N., Niu, G., & Sugiyama, M. (ICLR2023).

Takashi Ishida
(RIKEN/UTokyo)



Contents

1. Can we mitigate overfitting by avoiding too small training error?

2. Can we estimate the Bayes error accurately?

3. Summary

Formulation of Supervised Classification

- We are given input-output training data:

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y) \qquad \boldsymbol{x} \in \mathbb{R}^d, y \in \{1, \dots, c\}$$

- We want to obtain a classifier $g(\boldsymbol{x})$ that minimizes the test error: $R(g) = \mathbb{E}_{p(\boldsymbol{x},y)}[\ell\big(y, g(\boldsymbol{x})\big)]$

$\ell(y, \hat{y})$: Pointwise loss (e.g., cross-entropy)

- Since the true distribution is unknown, we minimize the training error in practice:

$$\hat{R}(g) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, g(\boldsymbol{x}_i))$$

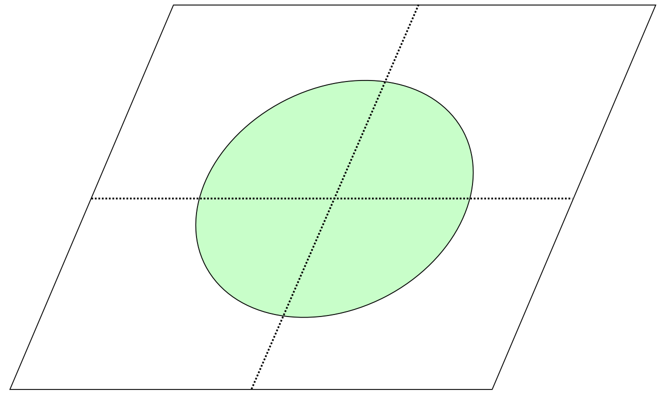
Coping with Overfitting

Regularization:

- Restrict the model complexity to avoid too small training error.

$$\hat{R}(g) + \lambda \cdot \Omega(g) \qquad \lambda \geq 0$$

Training error Regularizer Regularization parameter



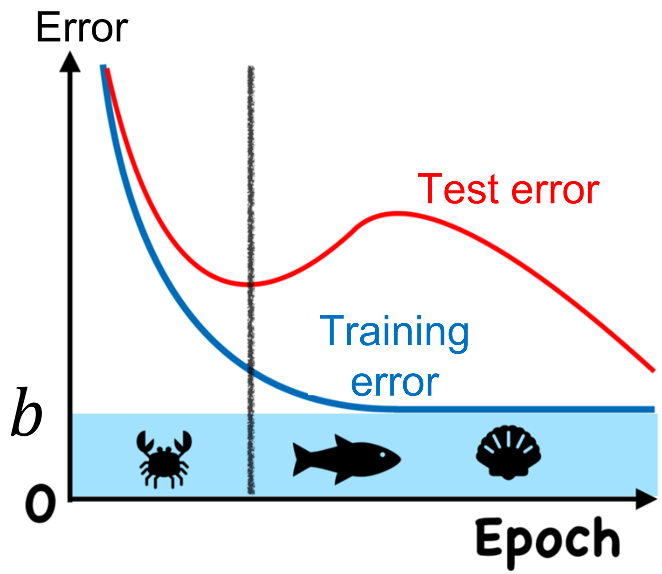
ℓ_2-regularization

Flooding: Ishida+ (ICML2020)

- Directly restrict the training error to be not too small.

$$|\hat{R}(g) - b| + b \qquad b \geq 0$$

Flood level



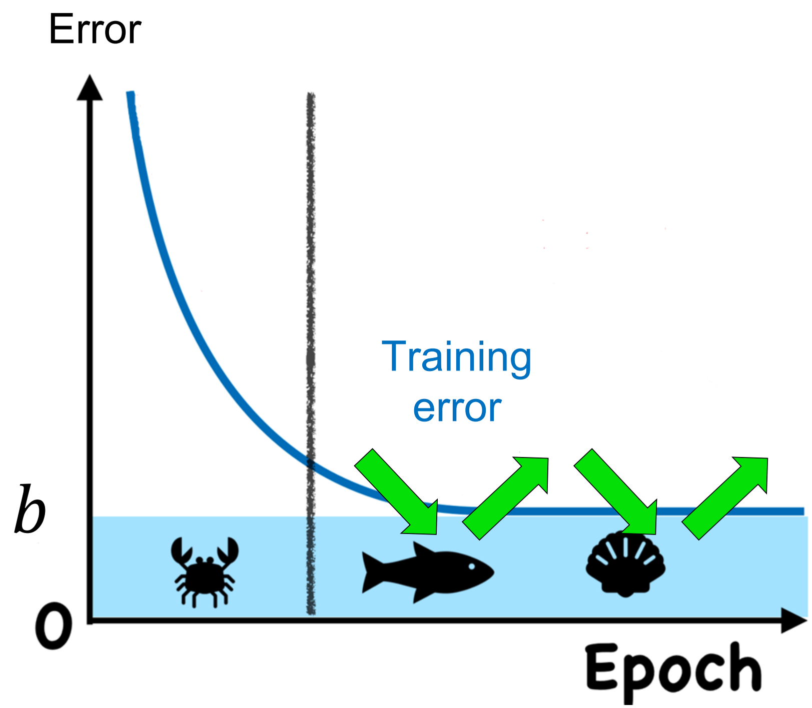
Behavior of Flooding

$$\tilde{R}(g) = |\hat{R}(g) - b| + b \qquad b \geq 0$$

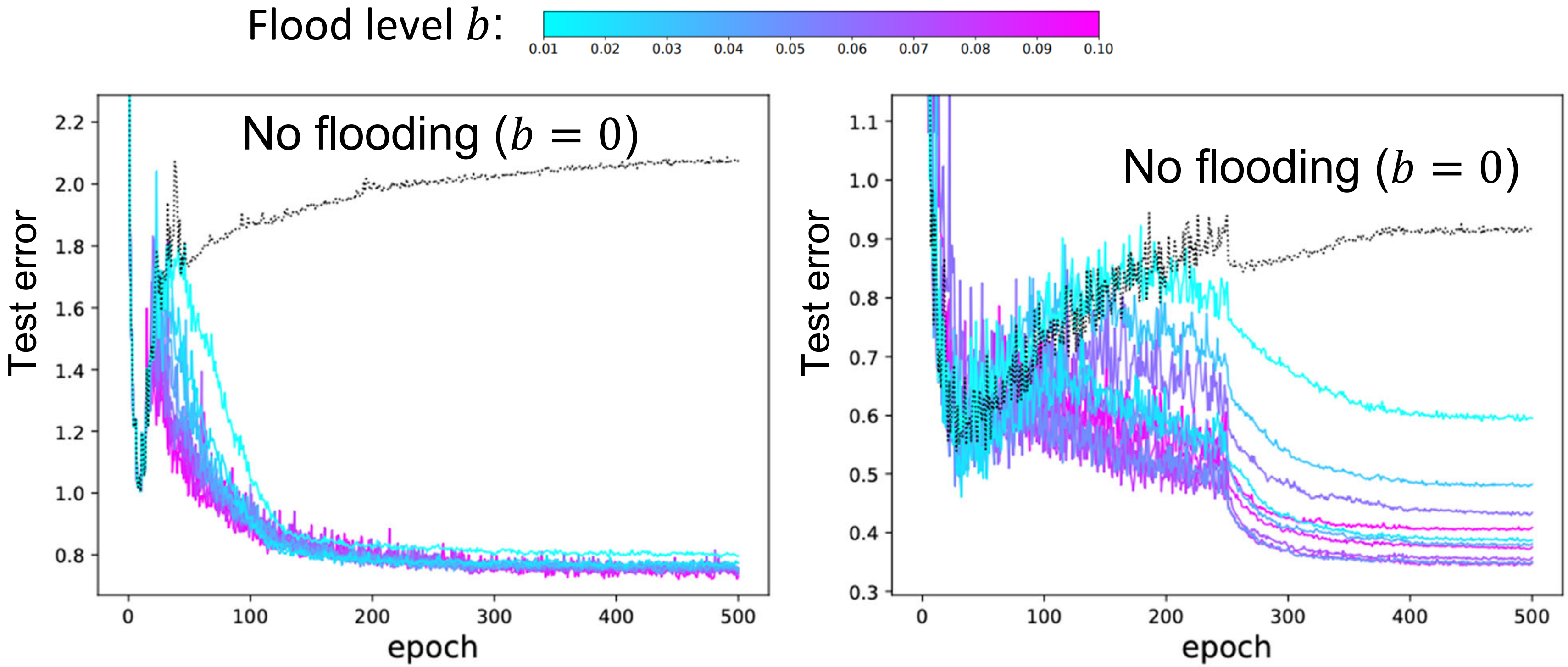
Training error

Flood level

- When $\hat{R}(g) \geq b$, $\tilde{R}(g) = \hat{R}(g)$:
  - Perform gradient descent.
- When $\hat{R}(g) < b$, $\tilde{R}(g) = -\hat{R}(g) + 2b$:
  - Perform gradient ascent.
- Therefore, when $\hat{R}(g) \approx b$,
  the solution does not stay,
  but is fluctuated (to find a better solution):
  - We treat b as a hyper-parameter.



Illustrative Experiments



CIFAR-10, ResNet44

CIFAR-10, ResNet44
Data augmentation & Learning rate decay

- With flooding, the test error improves.
- Flooding induces epoch-wise double descent for the test error.

Theoretical Justification

$$\tilde{R}(g) = |\hat{R}(g) - b| + b \qquad b \geq 0$$

Training loss

Flood level

- With proper choice of b, the mean squared error (MSE) of the flooded estimator $\tilde{R}$ is smaller than the original one R.
  - In practice, smaller b is safer.

Theorem 1. *Fix any measurable vector-valued function $\boldsymbol{g}$. If the flooding level b satisfies $\widehat{R}(\boldsymbol{g}) < b < R(\boldsymbol{g})$, we have*

$$\mathrm{MSE}(\widehat{R}(\boldsymbol{g})) > \mathrm{MSE}(\widetilde{R}(\boldsymbol{g})). \tag{10}$$

If $b \leq \widehat{R}(\boldsymbol{g})$, we have

$$\mathrm{MSE}(\widehat{R}(\boldsymbol{g})) = \mathrm{MSE}(\widetilde{R}(\boldsymbol{g})). \tag{11}$$

$$R(g) = \mathbb{E}_{p(\boldsymbol{x},y)}[\ell(y, g(\boldsymbol{x}))] \qquad \hat{R}(g) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, g(\boldsymbol{x}_i))$$

Experiments

		w/o early stopping		w/ early stopping	
Dataset	Model & Setup	w/o flood	w/ flood	w/o flood	w/ flood
MNIST	MLP	98.45%	**98.76%**	98.48%	**98.66%**
	MLP w/ weight decay	98.53%	**98.58%**	98.51%	**98.64%**
	MLP w/ batch normalization	98.60%	**98.72%**	**98.66%**	98.65%
Kuzushiji	MLP	92.27%	**93.15%**	92.24%	**92.90%**
	MLP w/ weight decay	92.21%	**92.53%**	92.24%	**93.15%**
	MLP w/ batch normalization	92.98%	**93.80%**	92.81%	**93.74%**
SVHN	ResNet18	92.38%	**92.78%**	92.41%	**92.79%**
	ResNet18 w/ weight decay	93.20%	–	92.99%	**93.42%**
CIFAR-10	ResNet44	**75.38%**	75.31%	74.98%	**75.52%**
	ResNet44 w/ data aug. & LR decay	88.05%	**89.61%**	88.06%	**89.48%**
CIFAR-100	ResNet44	**46.00%**	45.83%	**46.87%**	46.73%
	ResNet44 w/ data aug. & LR decay	63.38%	**63.70%**	63.24%	–

- Flooding significantly improves the prediction accuracy!

Contents

Bayes Error Estimation

- **Bayes error**: Minimum achievable test error.
  - Irreducible part of the test error!
- Why do we want to estimate it?
  - Investigate whether test-set overfitting occurs or not.
  - Use it for measuring task difficulty (e.g., acceptance/rejection decision at competitive conferences)

Best test error (March 2023)

MNIST	0.09%
CIFAR-10	0.50%
CIFAR-100	3.92%
ImageNet	8.90%

kwwsv=22sdshuvz lwkfrgh1frp 2vrwd

ICLR

ICML International Conference On Machine Learning

NEURAL INFORMATION PROCESSING SYSTEMS

kwwsv=22lfp o1ff2 kwwsv=22lfou1ff2 kwwsv=22qlsv1ff

Bayes Error Estimation

- **Naïve approach**:
  - With (big) supervised data, train a classifier.

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y) \quad \rightarrow \quad f(\boldsymbol{x})$$

$$\boldsymbol{x} \in \mathbb{R}^d, y \in \{1, \ldots, c\}$$

  - Use its validation error as an estimated Bayes error.

$$\{(\boldsymbol{x}'_i, y'_i)\}_{i=1}^{n'} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y) \quad \rightarrow \quad \frac{1}{n'}\sum_{i=1}^{n} \mathbf{1}[f(x'_i)=y'_i]$$

- **Drawback**:
  - Not accurate due to limited supervised training/validation data.
- **Our solution**: Ishida+ (ICLR2023)
  - Bayes error estimation without training a classifier.
  - We focus on binary classification (i.e., $y \in \{+1, -1\}$).

Bayes Error Estimation from Confidence Data

- Expression of the Bayes error β without $g(\boldsymbol{x})$:

$$\beta = \mathbb{E}_{p(\boldsymbol{x})}[\min\{p(y=+1|\boldsymbol{x}), p(y=-1|\boldsymbol{x})\}]$$

- Suppose we are given confidence data: $c_i = p(y=+1|\boldsymbol{x}_i)$

$\boldsymbol{x}_i \sim p(\boldsymbol{x})$

  - Our model-free and instance-free estimator:

$$\hat{\beta} = \frac{1}{n}\sum_{i=1}^{n}\min\{c_i, 1-c_i\}$$

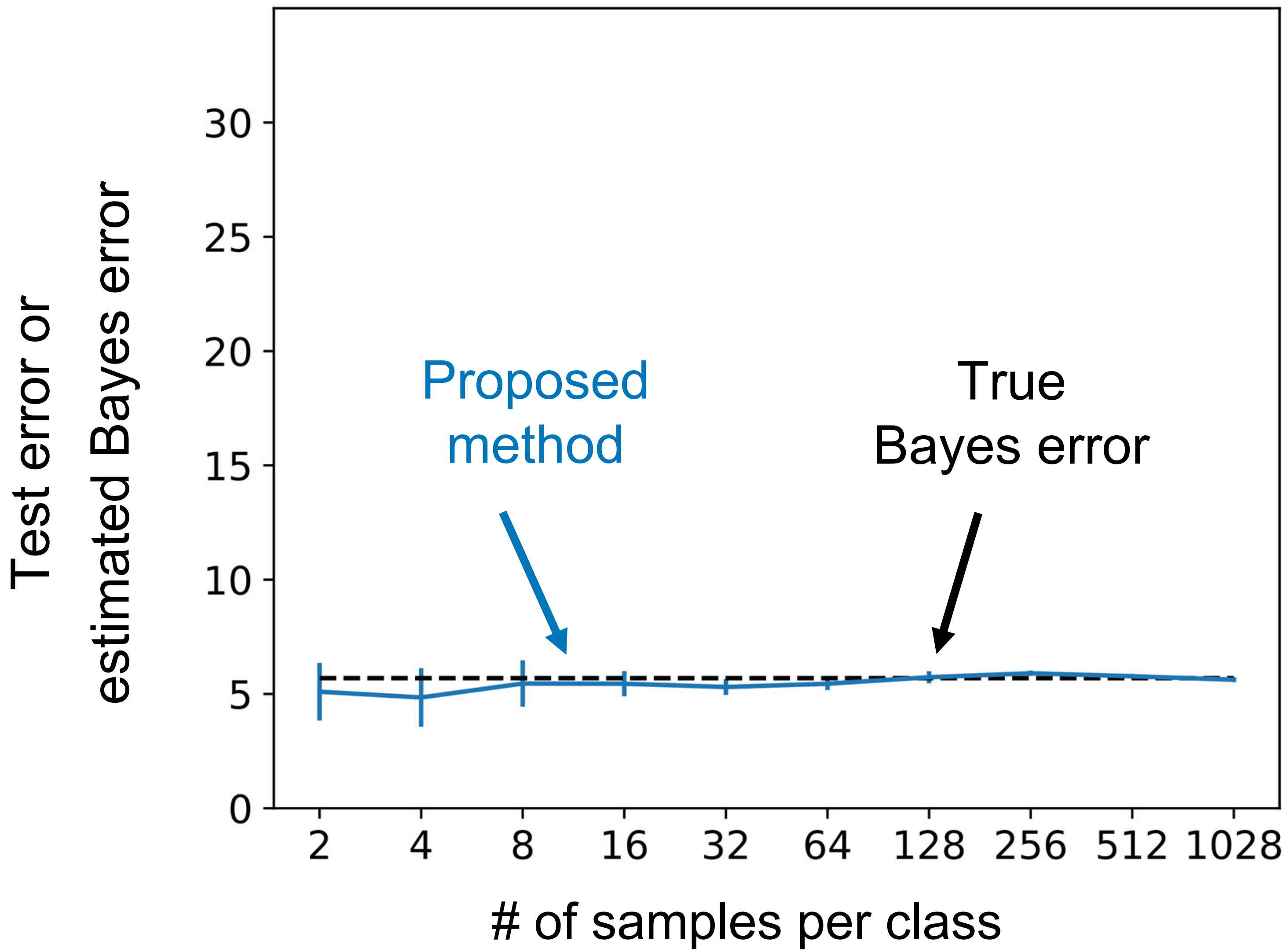
  - Unbiased and consistent:

$$\mathbb{E}[\hat{\beta}] = \beta \qquad |\hat{\beta} - \beta| \le \sqrt{\frac{1}{8n}\log\frac{2}{\delta}}$$

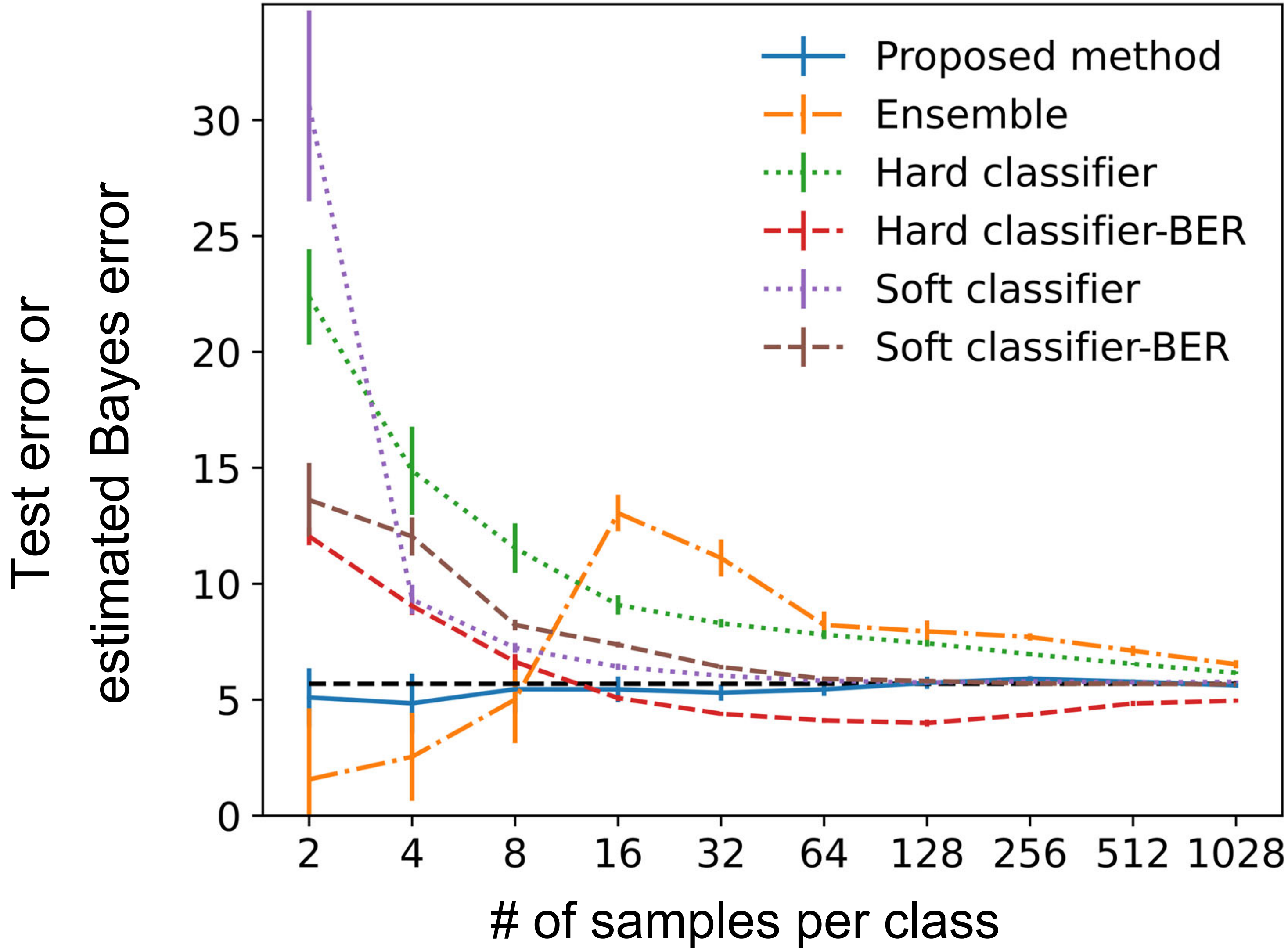
$\forall\delta > 0$

with probability $1-\delta$

Illustrative Example



Illustrative Example



Extension 1: Noisy Soft Labels and Sign Labels

- Suppose we are given
  - Noisy soft labels: $u_i = c_i + \xi_i$ $\quad \xi_i \sim \mathrm{truncN}(c_i, 0.4^2)\ \mathrm{s.t.}\ u_i \in [0,1]$
  - Sign labels: $s_i = \mathrm{sign}[c_i - 0.5]$ $\quad c_i = p(y = +1|\boldsymbol{x}_i)$ $\quad \boldsymbol{x}_i \sim p(\boldsymbol{x})$

$i = 1, \ldots, n$

- Proposed estimator:

$$\hat{\beta}_{\text{noisy}} = \frac{1}{n}\left(\sum_{i:s_i=+1}^{n} (1 - u_i) + \sum_{i:s_i=-1}^{n} u_i \right)$$

  - Unbiased and consistent:

$$\mathbb{E}[\hat{\beta}_{\text{noisy}}] = \beta \qquad |\hat{\beta}_{\text{noisy}} - \beta| \le \sqrt{\frac{1}{2n}\log\frac{2}{\delta}}$$

$\forall \delta > 0$

with probability $1 - \delta$

Extension 2: Multiple Hard Labels

- Suppose we are given
  - Multiple hard labels: $y_{i,j}$

$$y_{i,j} \overset{\text{i.i.d.}}{\sim} p(y|\boldsymbol{x}_i) \qquad \boldsymbol{x}_i \sim p(\boldsymbol{x})$$

$$i = 1, \dots, n, \; j = 1, \dots, m$$

- Proposed estimator:

$$\hat{\beta}_{\text{multi}} = \frac{1}{n} \sum_{i=1}^{n} \min\{v_i, 1 - v_i\} \qquad v_i = \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}[y_{i,j} = 1]$$

  - Asymptotically unbiased:

$$|\beta - \mathbb{E}[\hat{\beta}_{\text{multi}}]| \leq \frac{1}{2\sqrt{m}} + \sqrt{\frac{\log(2n\sqrt{m})}{2m}}$$

Extension 3: Positive Confidence

- Suppose we are given
  - Positive confidence: $r_i = p(y=+1|\boldsymbol{x}_i)$ $\quad \boldsymbol{x}_i \sim p(\boldsymbol{x}|y=+1)$

    $i = 1, \ldots, n_+$

- Proposed estimator: $\hat{\beta}_{\mathrm{Pconf}} = \pi_+ \left(1 - \frac{1}{n_+} \sum_{i=1}^{n_+} \max(0, 2 - \frac{1}{r_i})\right)$

  $\pi_+ = p(y=+1)$

  - Unbiased and consistent:

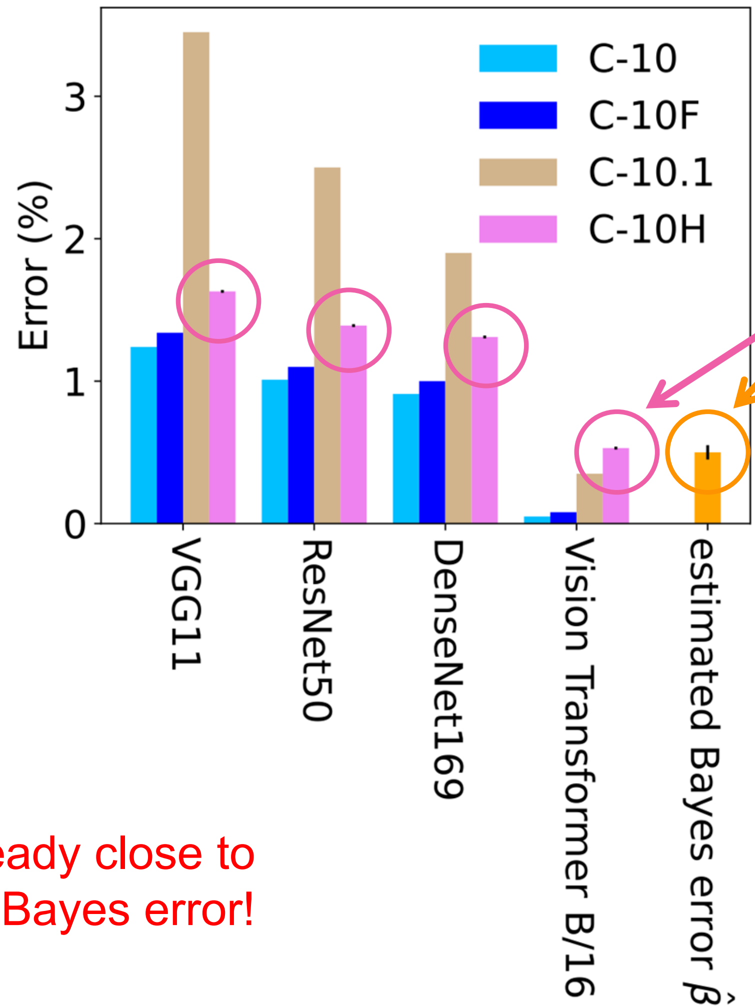
$$\mathbb{E}[\hat{\beta}_{\mathrm{Pconf}}] = \beta \qquad |\hat{\beta}_{\mathrm{Pconf}} - \beta| \leq \sqrt{\frac{\pi_+^2}{2n} \log \frac{2}{\delta}}$$

$\forall \delta > 0$

with probability $1 - \delta$

Benchmark Experiments

- CIFAR-10 (vs.)
  - Soft labels: class proportions of 50 human hard labels per image from CIFAR-10H (Peterson+ ICCV2019).



ViT's test error is already close to the Bayes error!

- Fashion-MNIST (vs.)
  - Multiple hard labels annotated by humans for each image from Fashion-MNIST.

ResNet18's test error	3.852% (± 0.041%)
Estimated Bayes error	3.478% (± 0.079%)

Already close to the Bayes error!

Difficulty of Paper Acceptance at ICLR

- Use the weighted average of ICLR reviewers' scores based on their confidence.

- Results:
  - The Bayes error is 6~10% (higher than CIFAR-10 and Fashion-MNIST).
  - No big changes over years.

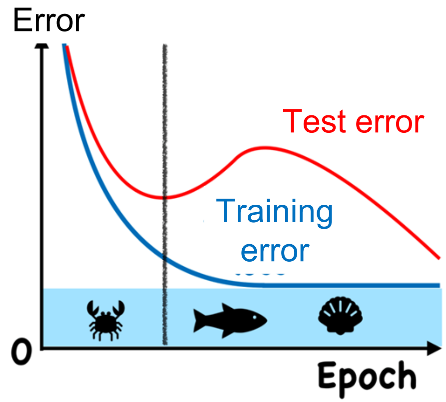
ICLR's Bayes error	
2017	6.8%(±1.0%)
2018	8.7%(±0.9%)
2019	7.9%(±0.7%)
2020	8.8%(±0.5%)
2021	9.3%(±0.5%)
2022	9.6%(±0.5%)
2023	8.0%(±0.4%)

- Demonstrates the benefit of our instance-free approach!

Contents

Summary

- **Flooding**: Keep the training error not too small. Ishida+ (ICML2020)
  - Instance-wise flooding. (Xie+ ICLR2022)
  - Instance-wise adaptive flooding. (Anonymous TMLR submitted)
  - Soft flooding. (Holland+ arXiv2023)
  - Time-series extension. (Cho+ NeurIPS2022)
  - Theoretical analysis. (Karakida+ ICML2023)



- **Bayes error estimation without explicit classifier training**: Ishida+ (ICLR2023)
  - Multi-class extension with clean soft labels. (Jeong+ NeurIPS2023)
  - Extension to the false positive rate. (Jeong+ arXiv2024)

$$\hat{\beta} = \frac{1}{n}\sum_{i=1}^{n} \min\{c_i, 1 - c_i\}$$

$c_i = p(y = +1|\boldsymbol{x}_i)$ $\boldsymbol{x}_i \sim p(\boldsymbol{x})$

- **Can we combine these two for auto-overfitting mitigation?**