Data Cleansing for Models Trained with SGD

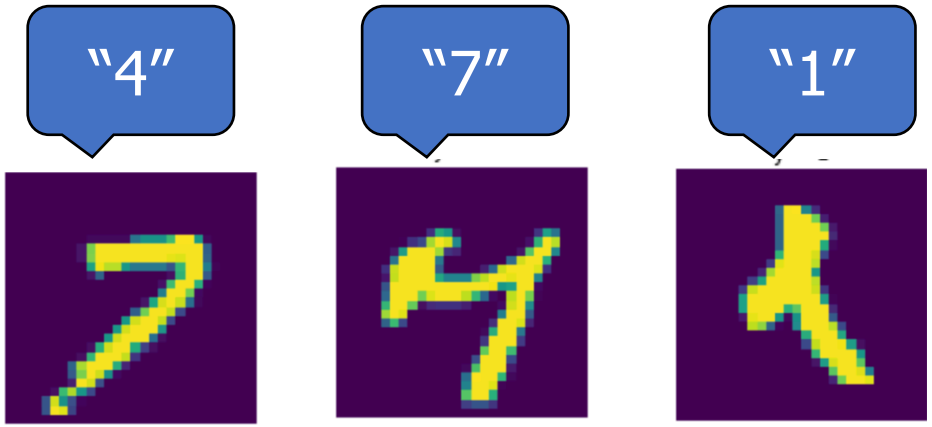
Satoshi Hara (Osaka Univ.)
Atsushi Nitanda (Tokyo Univ./RIKEN AIP)
Takanori Maehara (RIKEN AIP)

High-quality data is essential for ML.

- Low-quality data
  → Model with limited performance

Research Question

How can we improve the quality of data?



How can we improve the quality of data?

- Typical Approach
  - Data cleansing based on domain knowledge
  - Limitations
    - The workload of domain experts is limited.
      Impossible to check large data sets exhaustively.
    - Current domain knowledge may not be sufficient.
      There may be some causes the experts do not know.

Our Approach Automatic Data Cleansing

Identify "harmful" instances automatically.

Supervised Learning

- Training Data $D = \{z_n = (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}\}_{n=1}^N$

- Model $y = f(x; \theta)$ (θ: parameter of the model)
  - Examples

    Linear Model: $f(x; \theta) = \langle \theta, x \rangle$

    Deep Neural Network: $f(x; \theta)$

- Empirical Risk Minimization (ERM)

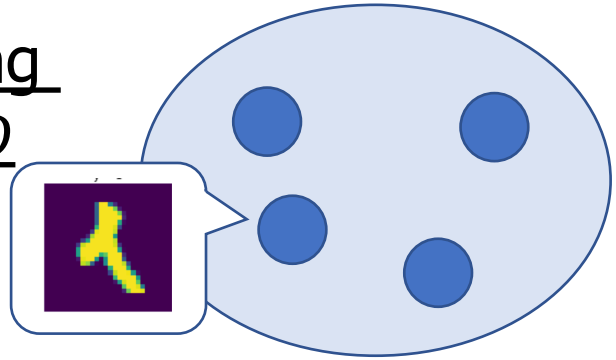
$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{z \in D} \ell(z; \theta)$$

Data Cleansing

[Koh&Liang, ICML'17]

- How largely the accuracy will change if one training instance is absent?

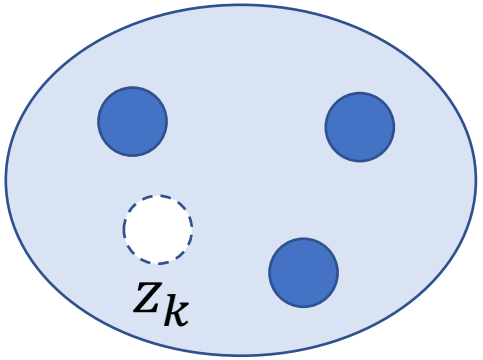
Training Data D



$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{z \in D} \ell(z;\theta)$$

Accuracy 98% !

Data with one instance absent $D \setminus \{z_k\}$



$$\hat{\theta}_{-k} = \underset{\theta}{\operatorname{argmin}} \sum_{z \in D \setminus \{z_k\}} \ell(z;\theta)$$

Accuracy 99% !!

Removing z_k leads to better accuracy.
→ z_k is "harmful".

Data Cleansing

[Koh&Liang, ICML'17]

Data Cleansing

- Remove one instance z_k from the training set D.
- Find z_k such that the model trained after the removal $\hat{\theta}_{-k}$ to have a better accuracy.

$$\hat{\theta} = \operatorname*{argmin}_{\theta} \sum_{z \in D} \ell(z; \theta)$$

$$\hat{\theta}_{-k} = \operatorname*{argmin}_{\theta} \sum_{z \in D \setminus \{z_k\}} \ell(z; \theta)$$

Which instance should be removed to improve the accuracy?

Data Cleansing

[Koh&Liang, ICML'17]

Def. Linear Influence

- For a given vector u, the linear influence r_k of the instance z_k is $r_k = \langle u, \hat{\theta}_{-k} - \hat{\theta} \rangle$.

- Good Model = Small Linear Influence
  - For the validation set D_V, let $u = \sum_{z \in D_V} \nabla_\theta \ell(z; \hat{\theta})$.
  - The first-order Taylor expansion yields

$$\langle u, \hat{\theta}_{-k} - \hat{\theta} \rangle \approx \sum_{z \in D_V} \left(\ell\left(z; \hat{\theta}_{-k}\right) - \ell\left(z; \hat{\theta}\right) \right)$$

  - Small validation loss ≈ Small linear influence

For data cleansing, we need to find an instance with a small linear influence.

Procedure of Data Cleansing

[Koh&Liang, ICML'17]

- Prepare
  - Training Data $D = \{z_n = (x_n, y_n)\}_{n=1}^{N}$
  - Trained Model $\hat{\theta}$
  - Vector u

for $k = 1, 2, \ldots, N$

Compute Linear Influence: $r_k = \langle u, \hat{\theta}_{-k} - \hat{\theta} \rangle$

Identify "harmful" instance: $\hat{k} = \underset{k}{\operatorname{argmin}}\, r_k$

Remove "harmful" instance: $D \leftarrow D \setminus \{z_{\hat{k}}\}$

Issue on Model Retraining

[Koh&Liang, ICML'17]

- We need $\hat{\theta}_{-k}$ to compute the linear influence.
  - $\hat{\theta}_{-k}$ can be obtained by retraining the model.

$$\hat{\theta}_{-k} = \operatorname*{argmin}_{\theta} \sum_{z \in D \setminus \{z_k\}} \ell(z; \theta)$$

Retraining is computationally very demanding.

Question

Can we estimate $\hat{\theta}_{-k} - \hat{\theta}$ without retraining?

Issue on Model Retraining

[Koh&Liang, ICML'17]

- We need $\hat{\theta}_{-k}$ to compute the linear influence.
  - $\hat{\theta}_{-k}$ can be obtained by retraining the model.

$$\hat{\theta}_{-k} = \operatorname*{argmin}_{\theta} \sum_{z \in D \setminus \{z_k\}} \ell(z; \theta)$$

Retraining is computationally very demanding.

Question

Can we estimate $\hat{\theta}_{-k} - \hat{\theta}$ without retraining?

- Prior Studies
  - Exact computation of $\hat{\theta}_{-k} - \hat{\theta}$
    - Linear Model + Squared Loss [Cook'77], GLM [Pregibon'81]
  - Approximation of $\hat{\theta}_{-k} - \hat{\theta}$
    - Strongly convex and smooth loss [Cook&Weisberg'80; Koh&Liang'17]

Limitations of Current Methods

- Current methods require two assumptions.
  - They do not hold true in current machine learning.

Assumption1 Convexity

- Loss function needs to be convex.
  - Current machine learning problems such as the training of deep learning are mostly non-convex.

Assumption2 Optimality

- An exact optimal solution of ERM needs to be obtained.
  - Current training algorithms such as SGD does not seek for an exact optima, but find an early-stopped solution.

This Study

Goal

Estimate $\hat{\theta}_{-k} - \hat{\theta}$ without restrictive assumptions.

- 1. Convexity → Valid even for non-convex losses.
- 2. Optimality → Valid even for non-optimal solutions.

Establish a valid estimation algorithm for current machine learning.

Overview of Our Approach

- SGD is de facto standard of current ML.
- Estimate $\hat{\theta}_{-k} - \hat{\theta}$ for the models trained with SGD.
  - Estimate $\hat{\theta}_{-k} - \hat{\theta}$ by "backpropating" through SGD.

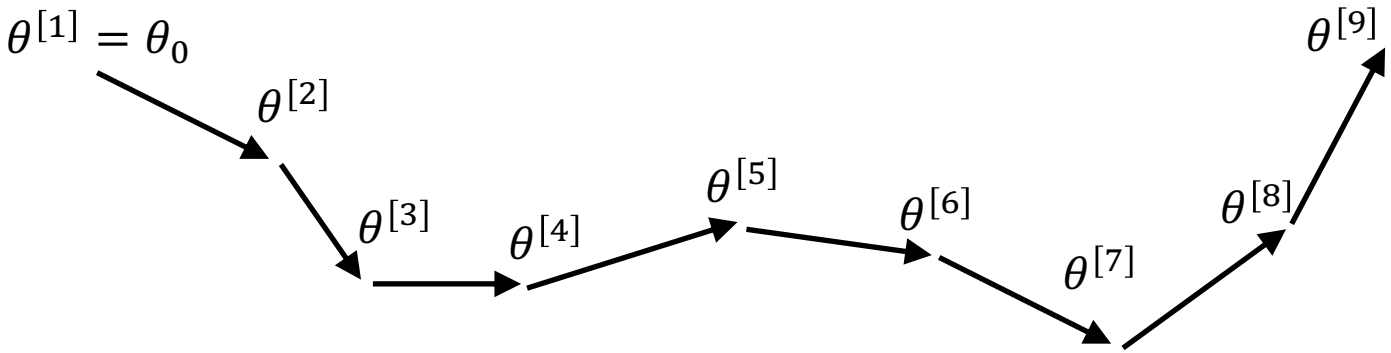
SGD

Mini-batch SGD

- Initialization: $\theta^{[1]} \leftarrow \theta_0$
- for $t = 1, 2, \ldots, T$

$$g_t \leftarrow \frac{1}{|S_t|} \sum_{z \in S_t} \nabla_\theta \ell(z; \theta^{[t]}) \qquad (S_t : \text{mini-batch})$$

$$\theta^{[t+1]} \leftarrow \theta^{[t]} - \eta_t g_t \qquad (\eta_t : \text{learning rate})$$



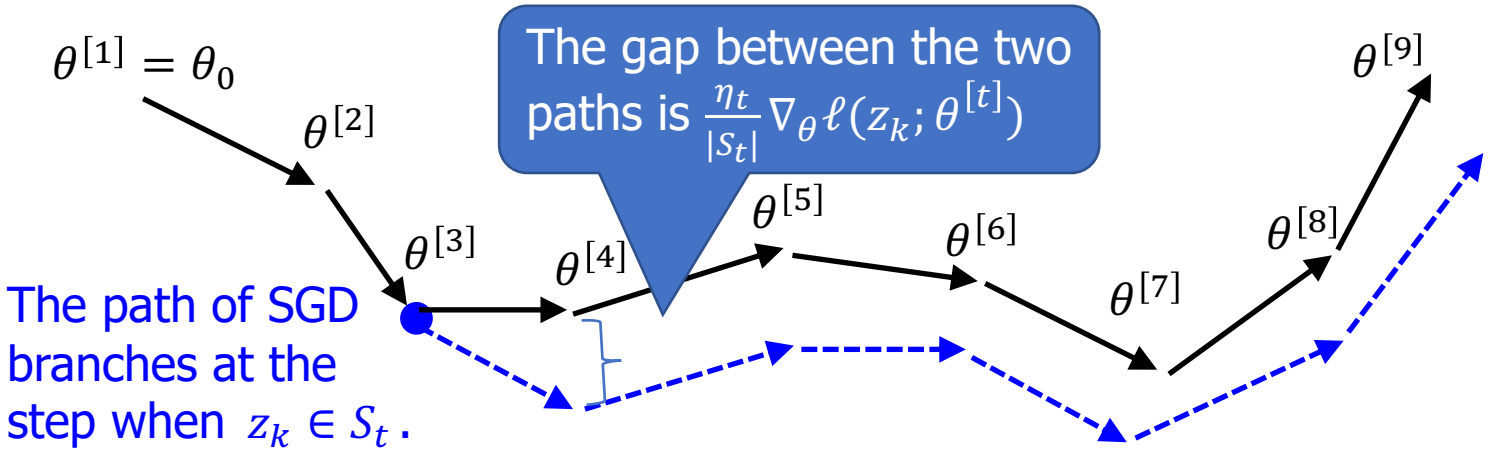
SGD

Counterfactual SGD (SGD without the k-th instance)

- Initialization: $\theta_{-k}^{[1]} \leftarrow \theta_0$
- for $t = 1, 2, \ldots, T$

$$g_{-k,t} \leftarrow \frac{1}{|S_t|} \sum_{z \in S_t \setminus \{z_k\}} \nabla_\theta \ell(z; \theta_{-k}^{[t]}) \quad (S_t : \text{mini-batch})$$

$$\theta_{-k}^{[t+1]} \leftarrow \theta_{-k}^{[t]} - \eta_t g_{-k,t} \quad (\eta_t : \text{learning rate})$$



The Proposed Estimator

- Consider One-epoch SGD for simplicity.
  - Each instance z_k is used only once.

The Proposed Estimator for $\theta_{-k}^{[T]} - \theta^{[T]}$

$$\Delta\theta_{-k} := \frac{\eta_{\pi(k)}}{|S_{\pi(k)}|} Z_{T-1} Z_{T-2} \cdots Z_{\pi(k)+1} \nabla_\theta \ell(z_k; \theta^{[\pi(k)]})$$

- $\pi(k)$: The step of SGD when z_k is used.
- $Z_t := I - \eta_t H_t$
- $H_t := \frac{1}{|S_t|} \sum_{z \in S_t} \nabla_\theta^2 \ell(z; \theta^{[t]})$

Hessian on the mini-batch S_t

Derivation of the Estimator

- SGD = A Specific "Feed Forward Neural Network"

$$\theta^{[T]} = h_{T-1} \circ h_{T-2} \circ \dots \circ h_{t+1} \circ h_t\left(\theta^{[t]}\right)$$

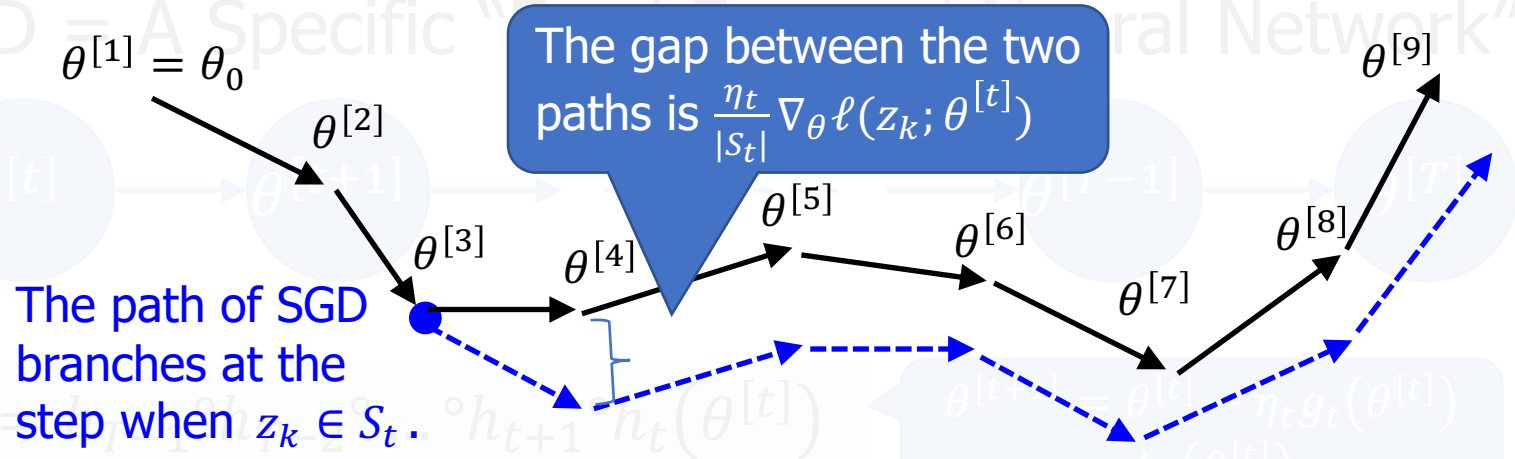
$$\theta^{[t+1]} = \theta^{[t]} - \eta_t g_t\left(\theta^{[t]}\right) =: h_t\left(\theta^{[t]}\right)$$

- Backpropagation

$$\theta_{-k}^{[T]} - \theta^{[T]} \approx \frac{\partial \theta^{[T]}}{\partial \theta^{[t+1]}}\left(\theta_{-k}^{[t+1]} - \theta^{[t+1]}\right)$$

$$= \frac{\partial h_{T-1}\left(\theta^{[T-1]}\right)}{\partial \theta^{[T-1]}} \frac{\partial h_{T-2}\left(\theta^{[T-2]}\right)}{\partial \theta^{[T-2]}} \dots \frac{\partial h_{t+1}\left(\theta^{[t+1]}\right)}{\partial \theta^{[t+1]}}\left(\theta_{-k}^{[t+1]} - \theta^{[t+1]}\right)$$

Derivation of the Estimator



- Backpropagation

$$\theta_{-k}^{[T]} - \theta^{[T]} \approx \frac{\partial \theta^{[T]}}{\partial \theta^{[t+1]}} \left(\theta_{-k}^{[t+1]} - \theta^{[t+1]} \right)$$

$$= \frac{\partial h_{T-1}\left(\theta^{[T-1]}\right)}{\partial \theta^{[T-1]}} \frac{\partial h_{T-2}\left(\theta^{[T-2]}\right)}{\partial \theta^{[T-2]}} \cdots \frac{\partial h_{t+1}\left(\theta^{[t+1]}\right)}{\partial \theta^{[t+1]}} \left(\theta_{-k}^{[t+1]} - \theta^{[t+1]} \right)$$

The gap between the two paths is $\frac{\eta_t}{|S_t|}\nabla_\theta \ell(z_k; \theta^{[t]})$

$$Z_{T-1} = I - \eta_{T-1} H_{T-1}$$

Derivation of the Estimator

The Proposed Estimator for $\theta_{-k}^{[T]} - \theta^{[T]}$

$$\Delta\theta_{-k} := \frac{\eta_{\pi(k)}}{|S_{\pi(k)}|} Z_{T-1} Z_{T-2} \cdots Z_{\pi(k)+1} \nabla_\theta \ell(z_k; \theta^{[\pi(k)]})$$

- $\pi(k)$: The step of SGD when z_k is used.

- Backpropagation

$$\theta_{-k}^{[T]} - \theta^{[T]} \approx \frac{\partial \theta^{[T]}}{\partial \theta^{[t+1]}} \left(\theta_{-k}^{[t+1]} - \theta^{[t+1]} \right)$$

$$= \frac{\partial h_{T-1}(\theta^{[T-1]})}{\partial \theta^{[T-1]}} \frac{\partial h_{T-2}(\theta^{[T-2]})}{\partial \theta^{[T-2]}} \cdots \frac{\partial h_{t+1}(\theta^{[t+1]})}{\partial \theta^{[t+1]}} \left(\theta_{-k}^{[t+1]} - \theta^{[t+1]} \right)$$

$$Z_{T-1} = I - \eta_{T-1} H_{T-1}$$

The gap between the two paths is $\frac{\eta_t}{|S_t|} \nabla_\theta \ell(z_k; \theta^{[t]})$

Estimation Error (Convex case)

- Assumption
  - The loss function is strongly convex, smooth, and twice differentiable.
  - $\exists \lambda, \Lambda > 0$ such that $\lambda I \preccurlyeq \nabla_\theta^2 \ell(z;\theta) \preccurlyeq \Lambda I$

Theorem

- For One-epoch SGD with each instance z_k used only once,

$$\left\| \left(\theta_{-k}^{[T]} - \theta^{[T]} \right) - \Delta\theta_{-k} \right\| \leq \sqrt{2(h_k(\lambda)^2 + h_k(\Lambda)^2)}$$

  - $h_k(a) \coloneqq \frac{\eta_{\pi(k)}}{|S_{\pi(k)}|} \prod_{t=\pi(k)+1}^{T-1} (1 - \eta_t a) \left\| \nabla_\theta \ell\left(z_k; \theta^{[\pi(k)]}\right) \right\|$

The estimator $\Delta\theta_{-k}$ and $\theta_{-k}^{[T]} - \theta^{[T]}$ are not very different.

Estimation Error (Non-Convex case)

- Assumption
  - The hessian of the loss function is Lipschitz.
    - $\left\|\nabla_\theta^2 \ell(z;\theta_1) - \nabla_\theta^2 \ell(z;\theta_2)\right\| \leq L\|\theta_1 - \theta_2\|$

Theorem

- Let $\|\nabla_\theta \ell(z;\theta)\| \leq G, \nabla_\theta^2 \ell(z;\theta) \preccurlyeq \Lambda I$, and $\eta = O(\gamma/\sqrt{T})$.
- For One-epoch SGD with each instance z_k used only once,

$$\left\|\left(\theta_{-k}^{[T]} - \theta^{[T]}\right) - \Delta\theta_{-k}\right\| \leq \frac{\gamma^2 T G^2 L}{\Lambda} \exp^{O(\gamma\Lambda\sqrt{T})}$$

The estimator $\Delta\theta_{-k}$ and $\theta_{-k}^{[T]} - \theta^{[T]}$ are not very different if the number of steps T is small.
In general, the error can grows exponentially.

Estimation of Linear Influence

- Preparation
  - Store the tuple $(S_t, \eta_t, \theta^{[t]})$ in each of the SGD step.

Algorithm Estimate $\hat{r}_k = \langle u, \Delta\theta_{-k} \rangle$ for all k

Initialization $\hat{r}_k \leftarrow 0, \forall k$

for $t = T-1, T-2, \ldots$ (Trace back SGD)

Read: $(S_t, \eta_t, \theta^{[t]})$

Update $\hat{r}_k$: $\hat{r}_k \leftarrow \hat{r}_k + \frac{\eta_t}{|S_t|} \left\langle u, \nabla_\theta \ell(z_k; \theta^{[t]}) \right\rangle, \forall z_k \in S_t$

Update u: $u \leftarrow u - \eta_t H_t u$ (Trace one step back for u)

Computationally Efficient Update of u

- The update $u \leftarrow u - \eta_t H_t u$ requires the Hessian H_t.
  - A naïve implementation requires $(\#\text{of parameters})^2$ memory to store the Hessian matrix.
  - This is not feasible for large models such as DNNs.

- We can update u without computing H_t explicitly.
  - We only need the matrix-vector product $\nabla_\theta^2 \ell\left(z; \theta^{[t]}\right) u$.
  - The following relationship enables us to compute the matrix-vector product efficiently.

$$\nabla_\theta^2 \ell\left(z; \theta^{[t]}\right) u = \nabla_\theta \left\langle \nabla_\theta \ell\left(z; \theta^{[t]}\right), u \right\rangle$$

Experiments

- Experiment1
  - Evaluate the estimation performance of the linear influence $\tilde{r}_k = \left\langle u, \theta_{-k}^{[T]} - \theta^{[T]} \right\rangle$.

- Experiment2
  - Evaluate the data cleansing performance.

Experiment1: Setups

Purpose

- Evaluate the estimation performance of the linear influence $\tilde{r}_k = \left\langle u, \theta_{-k}^{[T]} - \theta^{[T]} \right\rangle$.

- Datasets (Binary Classification)
  - Adult: High-income vs Low-income
  - 20Newsgroup: IBM article vs Apple article
  - MNIST: 1 vs 7

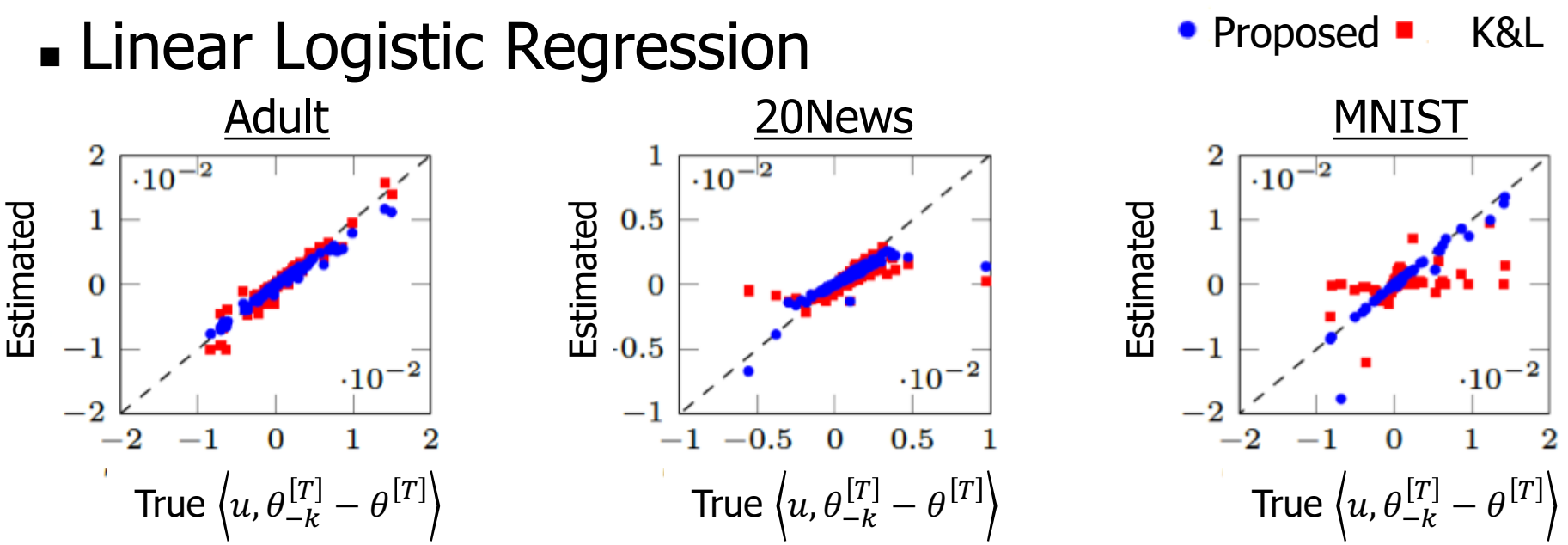
- Models
  - Linear logistic regression
  - Deep neural network (two layered, fully connected)

Experiment1: Procedure

- 1. Randomly select 200 instances for each of the training, the validation, and the test set.
- 2. Train a model with SGD.
- 3. Estimate the linear influence with the proposed method, and the method of K&L [Koh&Liang'17].
  - The vector u is constructed from the validation set.
    - $u = \sum_{z \in D_V} \nabla_\theta \ell(z; \hat{\theta})$, $\langle u, \theta_{-k} - \theta \rangle \approx \sum_{z \in D_V} \left(\ell(z; \theta_{-k}) - \ell(z; \theta) \right)$
- 4. Evaluate the estimation performance of the linear influence $\tilde{r}_k = \left\langle u, \theta_{-k}^{[T]} - \theta^{[T]} \right\rangle$.

Result of Experiment1

- Linear Logistic Regression

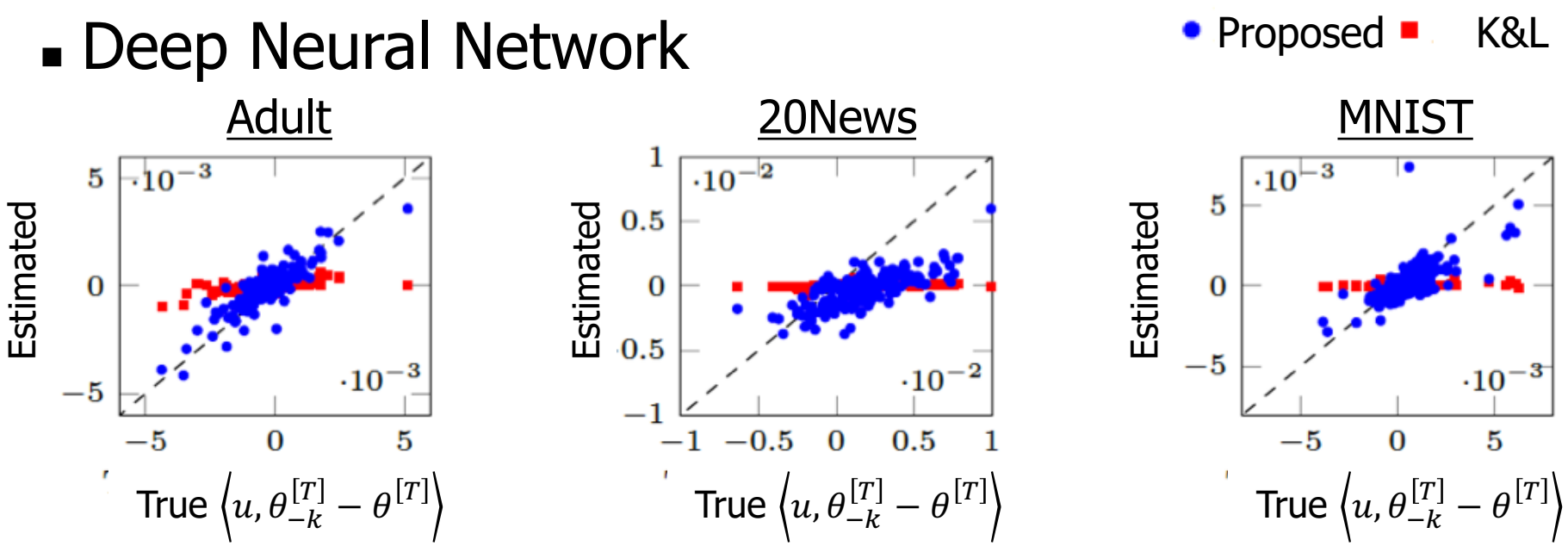


- Average of 30 runs ($\pm$ std.)

	Kendall's tau		Jaccard Index	
	Proposed	K&L	**Proposed**	K&L
Adult	**.93 (.02)**	.85 (.07)	**.80 (.10)**	.60 (.17)
20News	**.94 (.05)**	.82 (.15)	**.79 (.15)**	.52 (.19)
MNIST	**.95 (.02)**	.70 (.15)	**.83 (.10)**	.41 (.16)

Result of Experiment1

- Deep Neural Network



- Average of 30 runs ($\pm$ std.)

	Kendall's tau		Jaccard Index	
	Proposed	K&L	**Proposed**	K&L
Adult	**.75 (.10)**	.54 (.12)	**.59 (.16)**	.32 (.11)
20News	**.45 (.12)**	.37 (.12)	**.25 (.08)**	.11 (.07)
MNIST	**.45 (.12)**	.27 (.16)	**.37 (.15)**	.27 (.12)

Experiment2: Setups

Purpose

- Evaluate the data cleansing performance.
  - How largely the accuracy will improve by removing estimated “harmful” instances.

- Setups
  - Datasets: MNIST, CIFAR10
  - Model: CNN

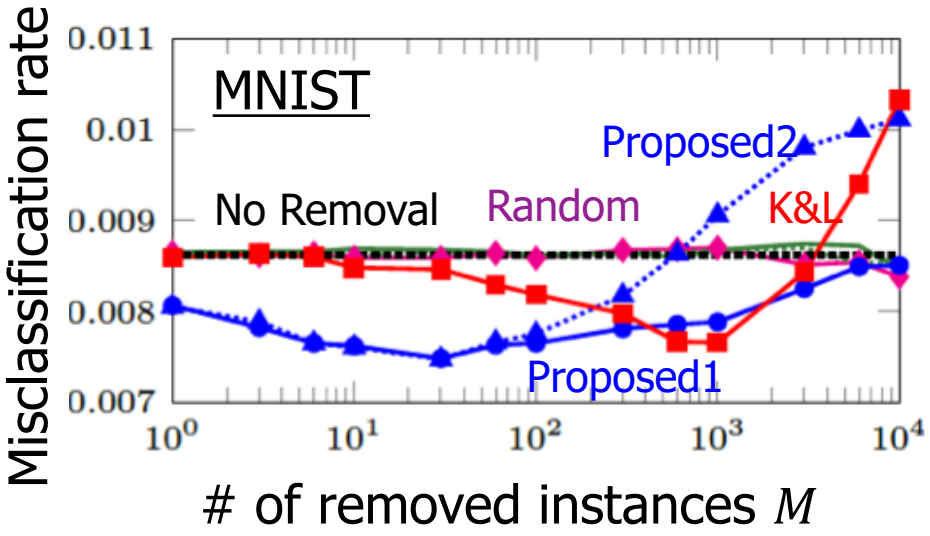
Experiment2: Procedure

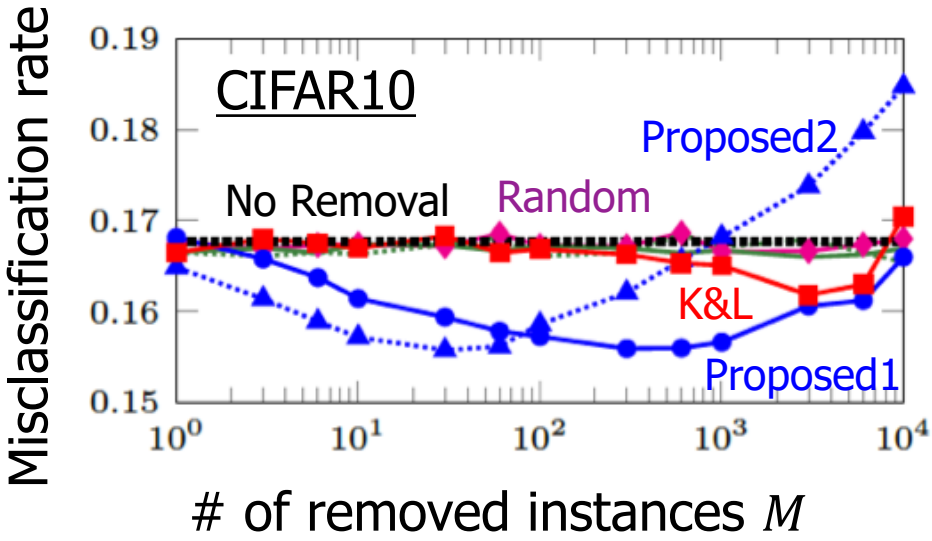
- 1. Randomly select 10,000 instances for each of the validation and the test set, and leave the rest to the training set.
- 2. Train a model with SGD.
- 3. Estimate the linear influence with the proposed method, and the method of K&L [Koh&Liang'17].
- 4. Evaluate how largely the test error has decreased after data cleansing.

Result of Experiment2

- Data cleansing
  - Remove M instances with small linear influences $\tilde{r}_k$ from the training set, and retrained the model.

Average test errors (average of 30 runs)

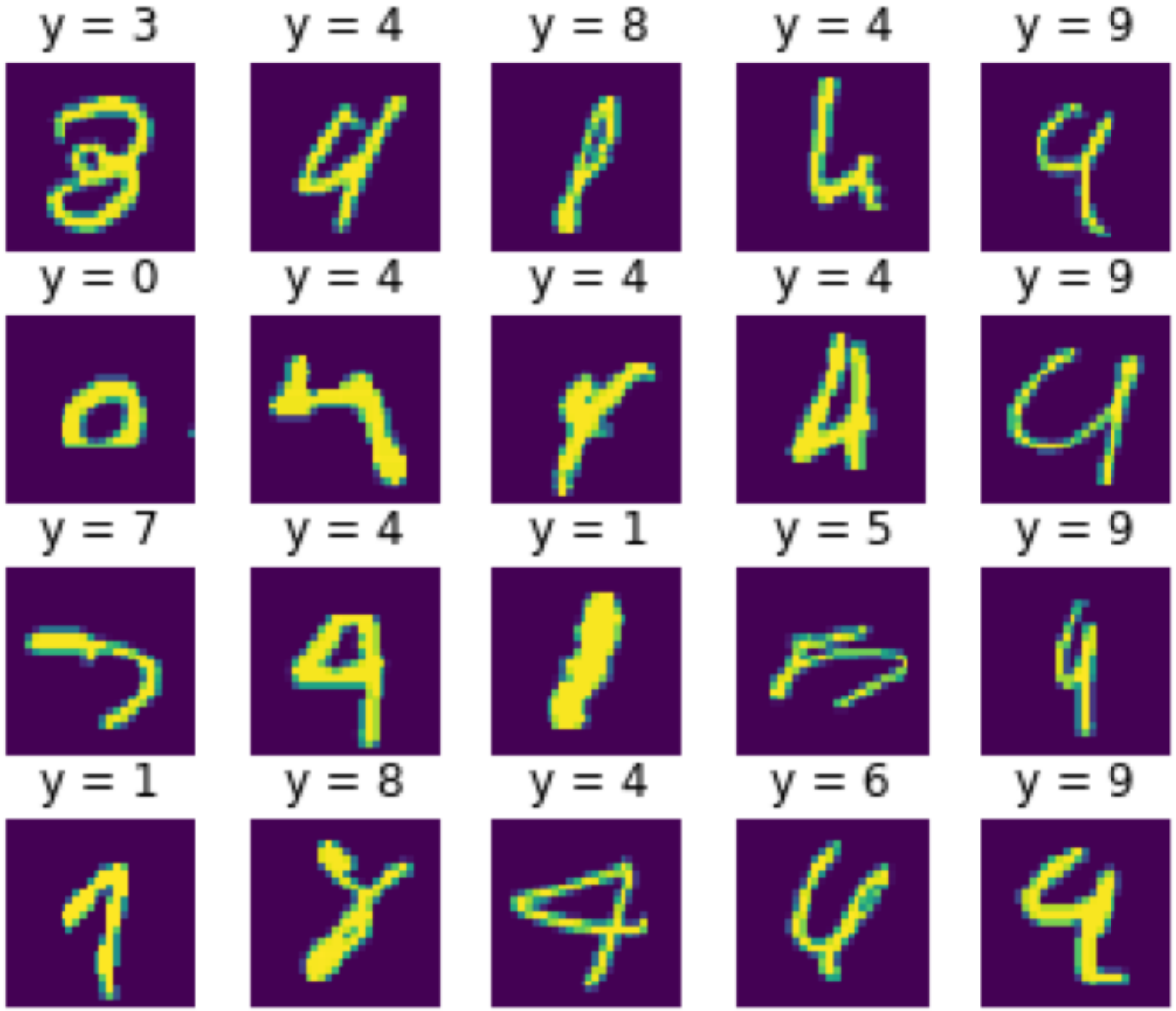




Proposed1	Proposed2
Trace back all the training epochs	Trace back only the last one epoch

Examples of Removed Instances

- MNIST: Top-20 instances



Examples of Removed Instances

- CIFAR10: Top-20 instances

Summary

- Automatic Data Cleansing
  - Remove some of the training instances so that the accuracy of the trained model to be improved without using any of the domain knowledge.

This Study

- Estimate "harmful" instances for models trained with SGD.
  - The proposed method does not require restrictive assumptions, convexity and optimality.
  - The proposed estimator is valid even for non-convex, non-optimal models.
- Demonstrated that it is possible to improve the model by removing estimated "harmful" instances.