TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

Less is Better: Unweighted Data Subsampling via Influence Function

Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, Shao-Lun Huang

2019/12/18

Data quality: Who is good?

Training set

Testing set

Label Y Image X_{tr}

Image X_{te}

Cat

Good or Bad?

Dog

Data quality: Noisy label

Training set | **Testing set**

Label Y | Image X_{tr} | Image X_{te}

Dog

Might be bad →

Dog

Data quality: Adversarial noise

Training set

Testing set

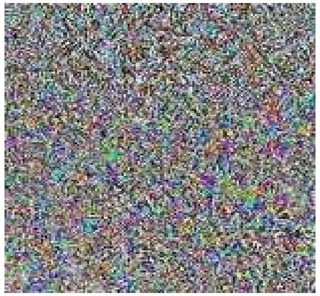
Label Y Image X_{tr}

Adversarial noise

Image X_{te}

Cat

$+\epsilon\ \times$



Might be bad

Dog

Data quality: Distribution shift

Training set

Testing set

Label Y Image X_{tr}

Image X'_{te}

Cat

Might be bad

Dog

Motivation

Label Y Image X_{tr} Image X_{te}

Cat  (1) $\phi(X_{tr}, X_{te})$ (2) $\pi(X_{tr})$ 

(1) Measuring data quality. How much a training data *influences* the model's prediction on testing data, i.e. $\phi(X_{tr}, X_{te})$

(2) Doing data selection. By which *probability* we select a data for training, namely probabilistic subsampling, considering data's quality, i.e. $\pi(X_{tr})$

Challenges

- **Dealing with ambiguity.** How to quantitatively define data's quality, in an unambiguous mathematical way, e.g. for text? Query?

- **Theoretical guidance.** How to build a theoretical reasonable method for data selection, and obtaining a better model via sub sample.

- **Robust selection.** How to ensure robustness of subsampling, considering controlling the performance on a set of distributions, i.e. $Q_{te} \in \{Q \mid D(Q||P) \leq \delta\}$

Leave-one-out (LOO) Training?

(1) Model $\hat{\theta}$ trained on *full* set (2) Test $\hat{\theta}$'s loss on each test image

N Training Images

$x_{tr}^{(0)}, y_{tr}^{(0)}$

⋮

$x_{tr}^{(N)}, y_{tr}^{(N)}$

$l\left(x_{te}^{(0)}; \hat{\theta}\right)$

⋮

$l\left(x_{te}^{(M)}; \hat{\theta}\right)$

M Testing Images

$x_{te}^{(0)}, y_{te}^{(0)}$

⋮

$x_{te}^{(M)}, y_{te}^{(M)}$

Leave-one-out (LOO) Training?

(3) Model $\tilde{\theta}$ trained on *sub* set

N-1 Training Images

$\cancel{x_{tr}^{(0)}, y_{tr}^{(0)}}$

$\vdots$

$x_{tr}^{(N)}, y_{tr}^{(N)}$

(4) Test $\tilde{\theta}$'s loss on each test image

$l\left(x_{te}^{(0)}; \tilde{\theta}\right)$

$\vdots$

$l\left(x_{te}^{(M)}; \tilde{\theta}\right)$

M Testing Images

$x_{te}^{(0)}, y_{te}^{(0)}$

$\vdots$

$x_{te}^{(M)}, y_{te}^{(M)}$

(5) Computing influence of $x_{tr}^{(0)}$

: I make test loss changes $\phi^{(0)} = \sum_{j=0}^{M} \left[l\left(x_{te}^{(j)}; \tilde{\theta}\right) - l\left(x_{te}^{(j)}; \hat{\theta}\right) \right]$

Approximate LOO via IF

Original objective function for training

$$\hat{\theta} = \mathrm{argmin}_{\theta} \frac{1}{N} \sum_{i=0}^{N} l(x_{tr}^{(i)}; \theta)$$

Reweight $l\left(x_{tr}^{(0)}; \theta\right)$ with a small ϵ

N Training Images

$x_{tr}^{(0)}, y_{tr}^{(0)}$

⋮

$x_{tr}^{(N)}, y_{tr}^{(N)}$

New objective function

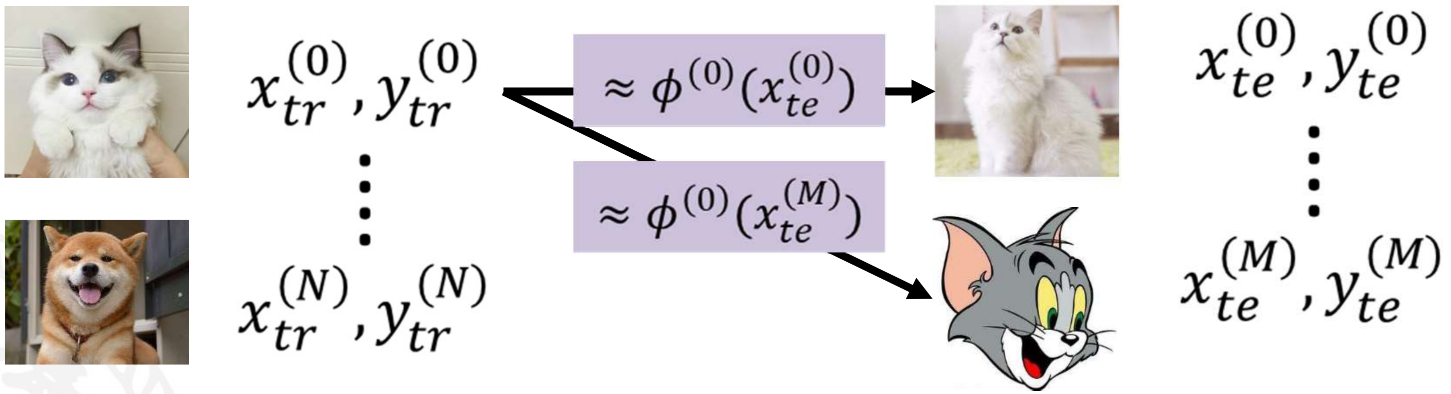
$$\hat{\theta}_{\epsilon} = \mathrm{argmin}_{\theta} \frac{1}{N} \sum_{i=0}^{N} l(x_{tr}^{(i)}; \theta) + \epsilon \times l\left(x_{tr}^{(0)}; \theta\right)$$

Note: we here assume $\epsilon \in \left[-\frac{1}{N}, 0\right]$ because

- If $\epsilon = -\frac{1}{N}$, the $l\left(x_{tr}^{(0)}; \theta\right)$ is removed
- If $\epsilon = 0$, the $l\left(x_{tr}^{(0)}; \theta\right)$ is kept

Approximate LOO via IF

N Training Images $\qquad$ M Testing Images



- Test loss change with new model $\hat{\theta}_\epsilon$ on each test sample (Koh & Liang, 2017):

$$l\left(x_{te}^{(j)}; \hat{\theta}_\epsilon\right) - l\left(x_{te}^{(j)}; \hat{\theta}\right) \approx \epsilon \times \phi^{(0)}\left(x_{te}^{(j)}\right) \quad \text{for all } j = 0,1,2 \ldots, M$$

- Definition of **Influence Function (IF)** $\phi^{(i)}\left(x_{te}^{(j)}\right)$:

$$\phi^{(i)}\left(x_{te}^{(j)}\right) \triangleq \left.\frac{\partial l\left(x_{te}^{(j)}, \theta_\epsilon\right)}{\partial \epsilon}\right|_{\epsilon=0} = -\nabla_\theta l(x_{te}, \hat{\theta}) H_{\hat{\theta}}^{-1} \nabla_\theta l(x_{tr}^{(i)}, \hat{\theta})$$

Koh, P. W., and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions In Proceedings of the 34th International Conference on Machine Learning-Volume 70, 1885–1894. JMLR. org.

How IF works

N Training Images

$x_{tr}^{(0)}, y_{tr}^{(0)}$

⋮

$x_{tr}^{(N)}, y_{tr}^{(N)}$

$\approx \phi^{(0)}(x_{te}^{(0)})$

$\approx \phi^{(0)}(x_{te}^{(M)})$

M Testing Images

$x_{te}^{(0)}, y_{te}^{(0)}$

⋮

$x_{te}^{(M)}, y_{te}^{(M)}$

Test loss change ΔL of all test samples:

$$\Delta L = \sum_{j=0}^{M} \left[l\left(x_{te}^{(j)}; \hat{\theta}_{\epsilon}\right) - l\left(x_{te}^{(j)}; \hat{\theta}\right) \right] \approx \epsilon \times \sum_{j=0}^{M} \phi^{(0)}\left(x_{te}^{(j)}\right) \triangleq \epsilon \times \Phi^{(0)}$$

$x_{tr}^{(0)}$'s influence

IF for data selection

Test loss change ΔL of all test samples:

$x_{tr}^{(0)}$'s influence

$$\Delta L = \sum_{j=0}^{M} \left[l\left(x_{te}^{(j)}; \hat{\theta}_{\epsilon}\right) - l\left(x_{te}^{(j)}; \hat{\theta}\right) \right] \approx \epsilon \times \sum_{j=0}^{M} \phi^{(0)}\left(x_{te}^{(j)}\right) \triangleq \epsilon \times \Phi^{(0)}$$

Let $\epsilon = -\frac{1}{N}$ ($x_{tr}^{(0)}$ is **dropped**), get $\Delta L \approx -\frac{1}{N} \times \Phi^{(0)}$, such that

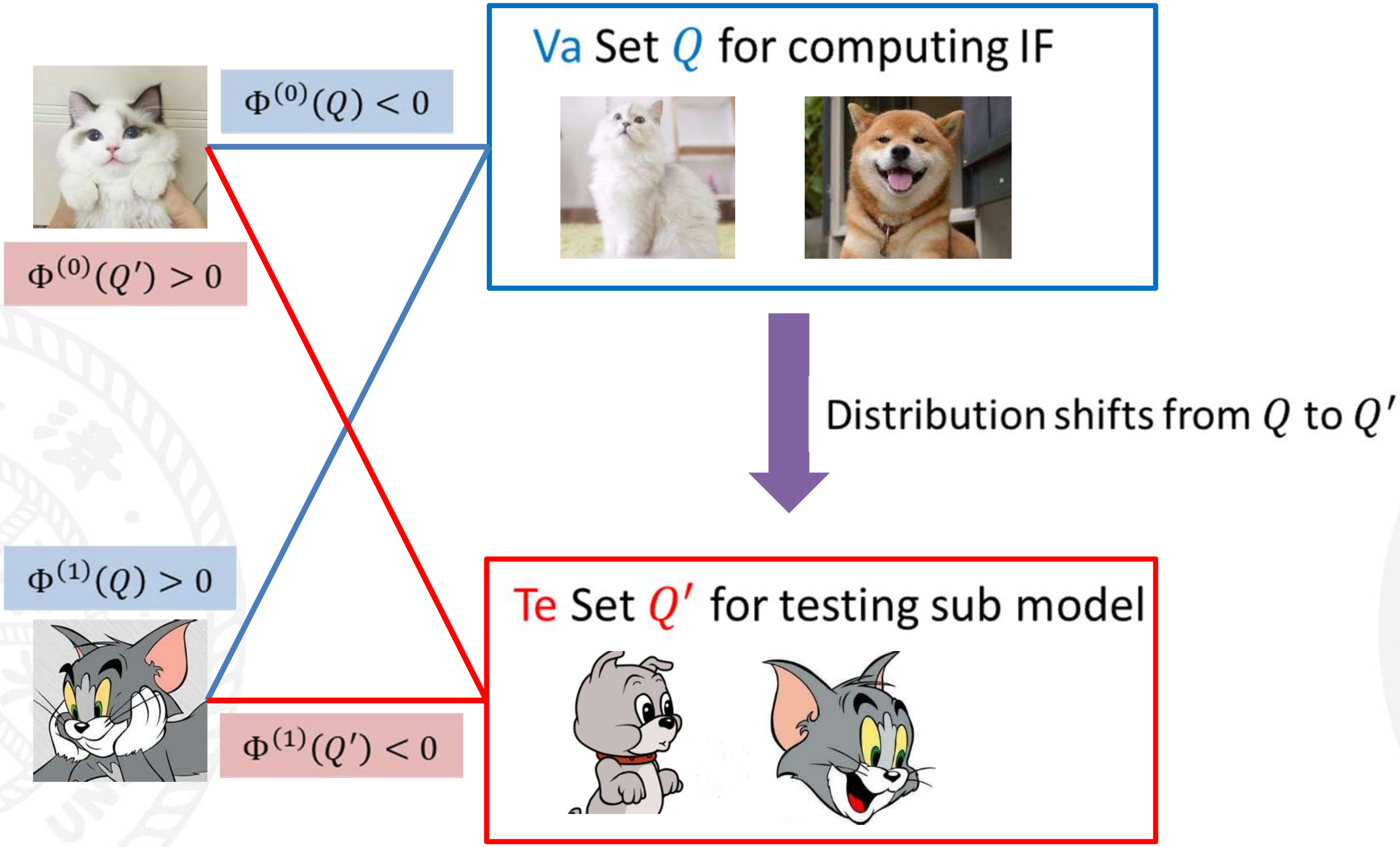
- If $\Delta L > 0$, then $\Phi^{(0)} < 0$

- **Dropping** $x_{tr}^{(0)}$ causes test loss **increasing**, we should keep it.

Q: Is it the whole story?

A: Not really !

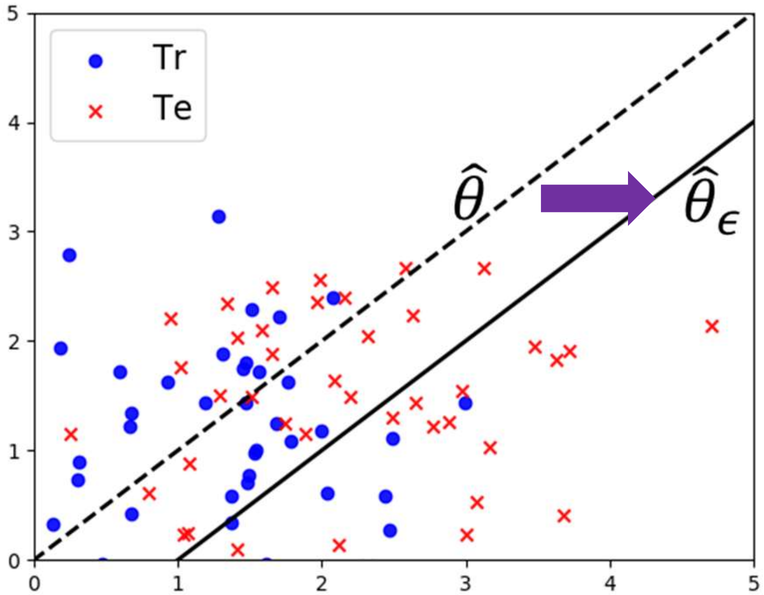
Cons: Over confidence



Overly confident on a single test set can undermine subsampling's robustness !

Set of distributions

- Define an uncertainty set of test distributions: $\mathcal{Q} \triangleq \left\{ Q \mid D_{\chi^2}(Q \parallel P) \leq \delta \right\}$
- Define the *worst-case* risk on $\mathcal{Q}$: $L\left(\mathcal{Q}; \hat{\theta}_\epsilon\right) \coloneqq \sup_{Q \in \mathcal{Q}} \left\{ \mathbb{E}_Q \left[l\left(\hat{\theta}_\epsilon; x\right) \right] \right\}$



Question:

If $\hat{\theta}$ goes to $\hat{\theta}_\epsilon$, how does $L\left(\mathcal{Q}; \hat{\theta}_\epsilon\right)$ change?

Worst-case risk with IF

清华-伯克利深圳学院
Tsinghua-Berkeley Shenzhen Institute

- Define an uncertainty set of test distributions: $\mathcal{Q} \triangleq \{ Q \mid D_{\chi^2}(Q \parallel P) \leq \delta \}$
- Define the *worst-case* risk on $\mathcal{Q}$: $L(\mathcal{Q}; \hat{\theta}_\epsilon) := \sup_{Q \in \mathcal{Q}} \{ \mathbb{E}_Q [l(\hat{\theta}_\epsilon; x)] \}$

Theorem *Let selection probability $\pi(\cdot)$ of a sample $x_{tr}^{(i)}$, takes its influence function $\Phi^{(i)}$ as input. If we have $\|\nabla \pi(\Phi^{(i)})\| \leq \sigma, \forall i = 1, \ldots, N$, then the worst-case risk $L(\mathcal{Q}; \hat{\theta}_\epsilon)$'s gradient norm has its upper bound:*

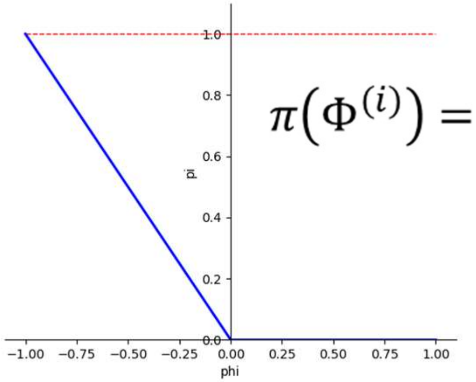
Guide to design sampling function

$$\|\nabla_\Phi L\| \leq \boxed{\sigma} \frac{\sqrt{2\delta + 1}}{N} \times \sqrt{\sum_{i=1}^{N} \Phi^{(i)}}$$

Design sampling functions

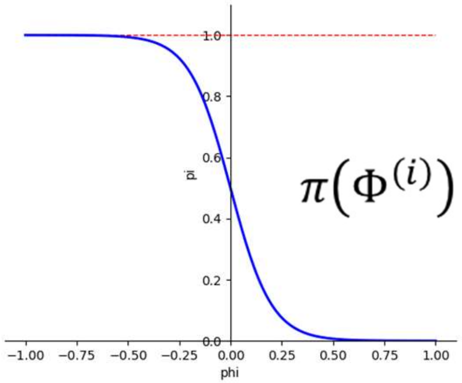
Probabilistic sampling

Linear sampling



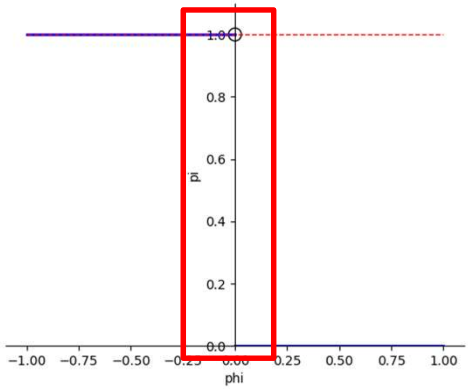
$$\pi\left(\Phi^{(i)}\right) = \max\left\{0, \min\left\{1, -\alpha\Phi^{(i)}\right\}\right\}$$

Sigmoid sampling



$$\pi\left(\Phi^{(i)}\right) = \mathrm{sigmoid}\left(\frac{-\alpha\Phi^{(i)}}{\max\Phi - \min\Phi}\right)$$

"*Over confidence*" sampling



Gap at zero point,
the $\|\nabla\pi\|$ is not bounded!

Data statistics

Table 1. Datasets Used in Experiments

Dataset	# samples	# features	Domain
UCI breast-cancer	683	10	Medical
diabetes	768	8	Medical
news20	19,954	1,355,192	Text
UCI Adult	32,561	123	Society
cifar10	60,000	3,072	Image
MNIST	70,000	784	Image
real-sim	72,309	20,958	Physics
SVHN	99,289	3,072	Image
skin non-skin	245,057	3	Image
criteo1%	456,674	1,000,000	CTR
covertype	581,012	54	Life
Avazu-app	14,596,137	1,000,000	CTR
Avazu-site	25,832,830	1,000,000	CTR
CPC	**>100M**	**>10M**	CTR

- From low dimension to high dimension
- From large data to small data

Results on Open Datasets

清华-伯克利深圳学院
TBSI Tsinghua-Berkeley Shenzhen Institute

Table 2. The average test loss with different sampling methods

Methods	Full set	Random	OptLR	Dropout	Linear(*)	Sigmoid(*)
UCI breast-cancer	0.0914	0.0944	0.0934	**0.0785**	**0.0873**	**0.0803**
diabetes	0.5170	0.5180	0.5232	**0.5083**	**0.5127**	**0.5068**
News20	0.5130	0.5177	0.5203	**0.5072**	**0.5100**	**0.5075**
UCI Adult	0.3383	0.3386	0.3549	0.3538	0.3384	**0.3382**
cifar10	0.6847	0.6861	0.7246	0.6851	**0.6822**	**0.6819**
MNIST	0.0245	0.0247	**0.0239**	**0.0223**	**0.0245**	**0.0231**
real-sim	0.2606	0.2668	0.2644	**0.2605**	0.2607	0.2609
SVHN	0.6129	**0.6128**	0.6757	0.6328	**0.6122**	**0.6128**
skin-nonskin	0.3527	**0.3526**	0.3529	0.4830	0.3713	**0.3527**
Criteo1%	0.4763	0.4768	0.4953	0.4786	**0.4755**	**0.4756**
Covertype	0.6936	**0.6933**	**0.6907**	0.7745	**0.6872**	**0.6876**
Avazu-app	0.3449	0.3449	0.3450	0.3576	**0.3446**	**0.3446**
Avazu-site	0.4499	0.4499	0.4505	0.5736	**0.4490**	**0.4486**
CPC	0.1955	0.1956	0.1958	0.1964	**0.1952**	**0.1953**

12 of 13 overtakes the *full set*, with selected *sub samples*.

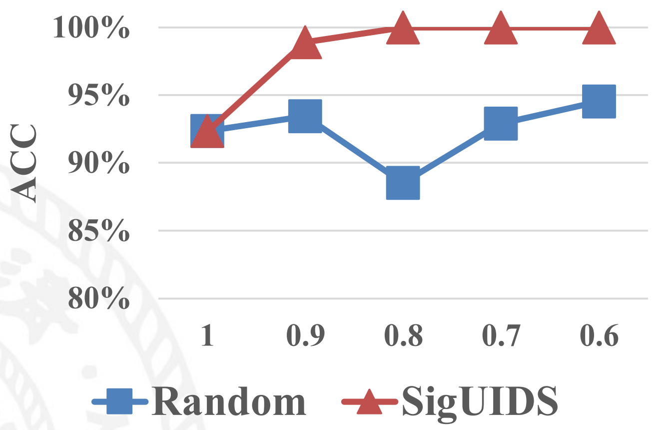
Sampling ratio: 95% from the full training set.

*: Ours methods.

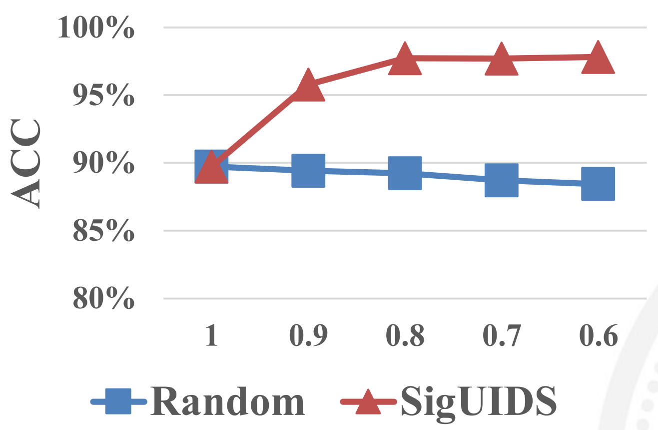
- The bold values indicate the loss smaller than the full set.
- The Dropout approach (Deterministic sampling) fails on many data sets.
- Our Linear and Sigmoid sampling methods overtake full set almost on each data set.

Noisy label setting

Putting 40% labels of training data flipped:



Diabetes data set



MNIST data set

- Under noisy data setting, our subsampling approaches show its enlarging superiority !

Conclusions

- Influence function is a useful measure for data's quality, which can guide us select good data and get better model.

- Our probabilistic sampling function can control the worst-case risk changes, to mitigate what is called *over confidence* problem.

References

1. Wang, Z., Zhu, H., Dong, Z., He, X., & Huang, S. L. (2019). Less Is Better: Unweighted Data Subsampling via Influence Function. arXiv preprint arXiv:1912.01321.
2. Wang, H., Zhu, R., & Ma, P. (2018). Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, *113*(522), 829-844.
3. Ai, M., Yu, J., Zhang, H., & Wang, H. (2018). Optimal Subsampling Algorithms for Big Data Generalized Linear Models. *arXiv preprint arXiv:1806.06761*.
4. Ting, D., & Brochu, E. (2018). Optimal Subsampling with Influence Functions. In *Advances in Neural Information Processing Systems, NIPS* (pp. 3650-3659).
5. Koh, P. W., & Liang, P. (2017, August). Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML-Volume 70*(pp. 1885-1894). JMLR. org.
6. Hsia, C. Y., Chiang, W. L., & Lin, C. J. (2018, November). Preconditioned conjugate gradient methods in truncated Newton frameworks for large-scale linear classification. In *Asian Conference on Machine Learning,ACML* (pp. 312-326).
7. Agarwal, N., Bullins, B., & Hazan, E. (2017). Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research*, *18*(1), 4148-4187.
8. Martens, J. (2010, June). Deep learning via Hessian-free optimization. In *ICML* (Vol. 27, pp. 735-742).
9. Wang, T., Huan, J., & Li, B. (2018). Data Dropout: Optimizing Training Data for Convolutional Neural Networks. 2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI), 39-46.
10. Ren, M., Zeng, W., Yang, B., & Urtasun, R. (2018). Learning to Reweight Examples for Robust Deep Learning. ArXiv, abs/1803.09050.
11. Cook, R. D., and Weisberg, S. 1980. Characterizations of an empirical influence function for detecting influential cases in regression. Technometrics 22(4):495–508.
12. Schnabel, T.; Swaminathan, A.; Singh,
13. A.; Chandak, N.; and Joachims, T. 2016. Recommendations as treatments: Debiasing learning and evaluation. arXiv preprint arXiv:1602.05352.

Thanks for Listening!

Q&A

Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, Shao-Lun Huang.

Less Is Better: Unweighted Data Subsampling via Influence Function.

AAAI 2020.