Less Is Better: Unweighted Data Subsampling via Influence Function

Zifeng Wang,[1*] **Hong Zhu,**[2†] **Zhenhua Dong,**[2‡] **Xiuqiang He,**[2] **Shao-Lun Huang**[1]

[1]Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, [2]Noah's Ark Lab, Huawei
Email: wangzf18@mails.tsinghua.edu.cn, {zhuhong8, dongzhenhua, hexiuqiang1}@huawei.com,
shaolun.huang@sz.tsinghua.edu.cn

Abstract

In the time of *Big Data*, training complex models on large-scale data sets is challenging, making it appealing to reduce data volume for saving computation resources by subsampling. Most previous works in subsampling are weighted methods designed to help the performance of subset-model approach the full-set-model, hence the weighted methods have no chance to acquire a subset-model that is better than the full-set-model. However, we question that *how can we achieve better model with less data?* In this work, we propose a novel Unweighted Influence Data Subsampling (UIDS) method, and prove that the subset-model acquired through our method can outperform the full-set-model. Besides, we show that overly confident on a given test set for sampling is common in Influence-based subsampling methods, which can eventually cause our subset-model's failure in out-of-sample test. To mitigate it, we develop a probabilistic sampling scheme to control the *worst-case risk* over all distributions close to the empirical distribution. The experiment results demonstrate our methods superiority over existed subsampling methods in diverse tasks, such as text classification, image classification, click-through prediction, etc.

Introduction

Bigger data can probably train a better model. It is almost the common sense nowadays for machine learning and deep learning practitioners. In the classical *Empirical Risk Minimization* (ERM) theory, it assumes that our training samples and test samples are i.i.d drawn from the same distribution. The learning theory tries to minimize the estimate of the generalization risk, namely the empirical risk. Therefore, when training samples are large enough, we can have hypothesis function working well on the test set by optimizing on the empirical risk.

However, the ERM faces several challenges: 1) a model is learned from training set which is generated by $P(x, y) = p_{train}(x)p(y|x)$, and the model is tested on $Q(x, y) = q_{test}(x)p(y|x)$. The *distribution shift* from P to Q violates the ERM's basic assumption; 2) unknown noise in data and its label is common in reality, causing some examples harmful for model's performance (Szegedy et al. 2014; Zhang et al. 2019); 3) training on large data sets imposes

significant burden on computation, some large-scale deep learning models require hundreds even thousands of GPUs.

Specifically, the *subsampling* approaches are initially proposed to cope with the last challenge. By virtue of subtle sampling regime, the selected subset can best approximate the original full set in terms of data distribution, hence the model can be trained on a compressed version of data set. In our work, we attempt to design sampling regime that can not only reduce the computation complexity, but also deal with several other ERM's difficulties. For instance, reweighting examples by sampling probabilities to fix mismatch between P and Q, and dropping noisy samples to strengthen the model's generalization ability.

Our work can be outlined in four points. First, instead of approaching the full-set-model $\hat{\theta}$, we prove that the model $\tilde{\theta}$ trained by a selected subset through our subsampling method can outperform the $\hat{\theta}$; second, we propose several probabilistic sampling functions, and analyze how the sampling function influences the worst-case risk (Bagnell 2005) changes over a χ^2-divergence ball. We further propose a surrogate metric to measure the confidence degree of the sampling methods over the observed distribution, which is useful for evaluating model's generalization ability on a set of distributions; third, for the sake of implementation efficiency, the Hessian-free mixed Preconditioned Conjugate Gradient (PCG) method is used to compute the influence function (IF) in sparse scenarios; last, complete experiments are conducted on diverse tasks to demonstrate our methods superiority over the existing state-of-the-art subsampling methods[1].

Related work

There are two main ideas to cope with the ERM challenges aforementioned: 1) pessimistic method that tries to learn model robust to noise or bad examples, including l_2-norm regularization, Adaboost (Freund and Schapire 1997), hard example mining (Malisiewicz, Gupta, and Efros 2011) and focal loss (Lin et al. 2017); and 2) optimistic method that modifies the input distribution directly. There are several genres of optimistic methods: the example reweighting method is used for dealing with distribution shift by (Bagnell 2005; Hu et al. 2016), and handling data bias by (Kumar, Packer, and Koller 2010; Ren et al. 2018); the sample selection method is applied to inspect and fix mislabeled data by

---

*Work done at Huawei Noah's Ark Lab as an intern.

†This author contributes equally with Zifeng Wang.

‡Corresponding author.

[1]The code can be found at https://github.com/RyanWangZf/Influence_Subsampling

(Zhang, Zhu, and Wright 2018). However, few of them have worked on alleviating the computational burden in terms of big data.

In order to reduce computation, the weighted subsampling method has been explored to approximate the maximum likelihood with a subset on the logistic regression (Fithian and Hastie 2014; Wang, Zhu, and Ma 2018), and on the generalized linear models (Ai et al. 2018). (Ting and Brochu 2018) introduces the IF in weighted subsampling for asymptotically optimal sampling probabilities for several generalized linear models. However, it is still an open problem about how to treat high variance of weight terms for weighted subsampling.

Specifically, the IF is defined by the Gateaux derivatives within the scope of *Robust Statistics* (Huber 2011), and extended to measure example-wise influence (Koh and Liang 2017) and feature-wise influence (Sliwinski, Strobel, and Zick 2019) on validation loss. The family of IF is mainly applied to design adversarial example and explain behaviour of black-box model previously. Recently, the IF on validation loss is used for targeting important samples, (Wang, Huan, and Li 2018) builds a sample selection scheme on Deep Convolutional Networks (CNN), and (Sharchilev et al. 2018) builds specific influential samples selection algorithm for Gradient Boosted Decision Trees (GBDT). However, by far there has no systematic theorem to guide IF's use in subsampling. Our work tries to build theoretical guidance for IF-based subsampling, which combines reweighting and subsampling together to synthetically cope with ERM's challenges, e.g. distribution shift and noisy data.

Preliminaries

Training samples $z_1, z_2, .., z_n \in \mathcal{X} \times \mathcal{Y}$ are generated from $P(x, y)$, and $x \in \mathbb{R}^d$ where the d is the number of feature dimension. Specifically for classification task, we have hypothesis function $h_\theta \in \mathcal{H} : \mathbb{R}^d \to \mathbb{R}$ parameterized by $\theta \in \Theta$. The goal is to minimize the 0-1 *risk* $\mathcal{R}_\theta(P) = \mathbb{E}_P[\mathbb{I}(h_\theta(x) \neq y)]$, and learn the optimal $\hat{\theta}$. For computational tractability, researchers focus on minimization of the *surrogate loss*, e.g. the log loss for binary classification:

$$l(z, h_\theta) = -y \log h_\theta(x) - (1 - y) \log(1 - h_\theta(x)) \quad (1)$$

Therefore, the risk minimization problem can be empirically approximated by $\hat{\theta} \triangleq \arg\min_\theta \frac{1}{n} \sum_{i=1}^n l(z_i, h_\theta)$. For convenience, we denote $l(z_i, h_\theta)$ by $l_i(\theta)$. The main notations are listed in Table 1.

Weighted subsampling. For a general subsampling framework, each sample is assigned with a random variable $O_i \sim Bern(\pi_i)$, indicating whether this sample is selected or not, such that $[O_i = 1] \leftrightarrow [z_i \text{ is selected}]$. The weighted subsampling methods have the similar form of objective function $\mathcal{R}_w$ on the subset:

$$\mathcal{R}_w = \frac{1}{n} \sum_{i, O_i=1}^n \frac{1}{\pi_i} l_i(\theta) \quad (2)$$

where each term is assigned the inversion of its sampling probability $1/\pi_i$ as the weight. It is similar to the technique

Table 1: Main notation.

z_i, z_j	training and testing sample, $\in \mathbb{R}^d \times \mathbb{R}$.
$\hat{\theta}, \tilde{\theta}(\text{or } \hat{\theta}_\epsilon)$	full-set-model and subset-model, $\in \mathbb{R}^d$.
$\phi_i(\hat{\theta})$	z_i's influence of whole test set risk, $\in \mathbb{R}$.
$\psi_\theta(z_i)$	z_i's influence of model parameter, $\in \mathbb{R}^d$.
ϵ_i	perturbation put on z_i's loss term, $\in \mathbb{R}$.
π_i	sampling probability of z_i, $\in \mathbb{R}^+$.
$l_i(\theta)$	model θ's risk on training sample z_i, $\in \mathbb{R}$.
$l_j(\theta)$	model θ's risk on test sample z_j, $\in \mathbb{R}$.
P	training distribution.
Q'	a specific test distribution.

used in *Causal Inference* to handle the selection bias (Schnabel et al. 2016), the Eq.(3) derives the expectation of $\mathcal{R}_w$ on O:

$$\mathbb{E}_O(\mathcal{R}_w) = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{O_i}(O_i \times \frac{1}{\pi_i} l_i(\theta)) = \frac{1}{n} \sum_{i=1}^n l_i(\theta) \quad (3)$$

The expectation of the $\mathcal{R}_w$ on a subset is the same as the empirical risk on the full set, which means the weighted subsampling methods aim at finding optimal $\{\pi_i\}_{i=1}^n$ to let the subset risk minimizer $\tilde{\theta}$ as close to the full-set risk minimizer $\hat{\theta}$ as possible.

This weighted approach has three main challenges: 1) we need to modify the existing training procedures to accommodate the weighted loss function; 2) because π can be small and its inversion ranges widely, the weighted loss function suffers from high variance; and 3) most importantly, as the weighted methods build a consistent estimator of the full-set-model (Ai et al. 2018), it theoretically assumes that subset-model cannot outperform the full-set-model.

Unweighted subsampling. We propose a novel *Unweighted* subsampling method which does not require $\frac{1}{\pi_i}$ in its objective function:

$$\mathcal{R}_{uw} = \frac{1}{|\{i, O_i = 1\}|} \sum_{i, O_i=1} l_i(\theta) \quad (4)$$

where the $|\cdot|$ means cardinality of the set. From Eq.(4), we can find the expectation of subset-model's risk is no longer equal to the full-set-model's. This formula can be seen as reweighting the samples with respect to their sampling probabilities implicitly. It directly overcomes the first two challenges in weighted subsampling above-mentioned, but further efforts are required to solve the last one, which will be introduced in the following section. An intuitive demonstration of the difference between the weighted and unweighted methods is shown as Fig. 1.

Influence functions. If the i'th training sample is upweighted by a small ϵ' corresponding to its loss term, then the perturbed risk minimizer is

$$\hat{\theta}_{\epsilon'} \triangleq \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n l_i(\theta) + \epsilon' l_{i'}(\theta) \quad (5)$$
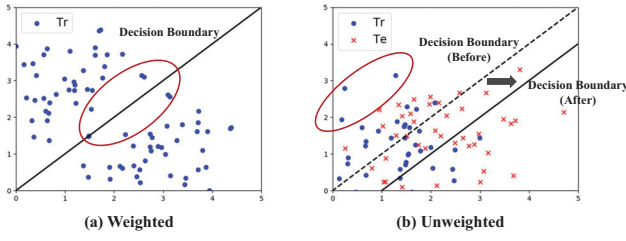
(a) Weighted   (b) Unweighted

Figure 1: (a) if the blue points (training samples) within the red circle are removed, the new optimal decision boundary is still same as the former one; (b) if removing blue points in the red circle, the new decision boundary shifts from the left, while achieves better performance on the Te set.

The basic idea of IF is to approximate the change of parameter θ (Cook and Weisberg 1980):

$$\psi_\theta(z_i) \triangleq \left. \frac{d\hat{\theta}_\epsilon}{d\epsilon} \right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_\theta l_i(\hat{\theta}) \qquad (6)$$

or test risk on a given test distribution Q' from (Koh and Liang 2017):

$$\phi(z_i \sim P, z_j \sim Q') \triangleq \left. \frac{dl_j(\hat{\theta}_\epsilon)}{d\epsilon} \right|_{\epsilon=0}$$
$$= -\nabla_\theta l_j(\hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta l_i(\hat{\theta}) \qquad (7)$$

where we have z_i, z_j from training and test set, and the $H_{\hat{\theta}} \triangleq \frac{1}{n} \sum_{i=1}^n \nabla_\theta^2 l_i(\hat{\theta})$ is the Hessian matrix based on full set risk minimizer $\hat{\theta}$, which is positive definite (PD) if the empirical risk is twice-differentiable and strictly convex in θ.

Methodology

The key challenge lies on that the $\hat{\theta}$ may not be the best risk minimizer corresponding to Q', due to the distribution shift between P and Q' and unknown noisy samples in the training set. With the advent of IF, we can measure one sample's influence on test distribution without prohibitive leave-one-out training. The essential philosophy here is, given a test distribution Q', some samples in training set cause increasing test risk. If they are downweighted, accordingly we can have less test risk than before, namely $\mathcal{R}_{\tilde{\theta}}(Q') \leq \mathcal{R}_{\hat{\theta}}(Q')$ where the $\tilde{\theta}$ is the new model learned after some harmful samples are downweighted.

Subset can be better

Considering there are n perturbations put on each training sample: $\vec{\epsilon} = (\epsilon_1, \epsilon_2, ..., \epsilon_n)^\top$, the perturbed risk minimizer is denoted as $\hat{\theta}_\epsilon$. Given m samples $\{z_j\}_{j=1}^m$ from another distribution Q', the objective is to design the $\vec{\epsilon}$ that minimizes the test risk $\mathcal{R}_{\hat{\theta}_\epsilon}(Q') = \frac{1}{m} \sum_{j=1}^m l_j(\hat{\theta}_\epsilon)$.

According to the definition of IF in Eq. (7), we can approximate the loss change of $z_j \sim Q'$ if $z_i \sim P$ is upweighed by a ϵ_i:

$$l_j(\hat{\theta}_{\epsilon_i}) - l_j(\hat{\theta}) \approx \epsilon_i \times \phi(z_i, z_j, \hat{\theta}) \qquad (8)$$

which can be extended to the whole test distribution as following:

$$\mathcal{R}_{\hat{\theta}_{\epsilon_i}}(Q') - \mathcal{R}_{\hat{\theta}}(Q') = \frac{1}{m} \sum_{j=1}^m (l_j(\hat{\theta}_{\epsilon_i}) - l_j(\hat{\theta}))$$
$$\approx \epsilon_i \times \frac{1}{m} \sum_{j=1}^m \phi(z_i, z_j, \hat{\theta}) \qquad (9)$$

For convenience, we use $\phi_i(\hat{\theta}) \triangleq \sum_{j=1}^m \phi(z_i, z_j, \hat{\theta})$ to indicate one training sample z_i's influence over whole Q'. Therefore, with the $\hat{\theta}_\epsilon$ under the perturbation of $\vec{\epsilon}$, the test risk change can be approximated as following[2]:

$$\mathcal{R}_{\hat{\theta}_\epsilon}(Q') - \mathcal{R}_{\hat{\theta}}(Q') = \frac{1}{m} \sum_{i=1}^n \sum_{j=1}^m (l_j(\hat{\theta}_{\epsilon_i}) - l_j(\hat{\theta}))$$
$$\approx \frac{1}{m} \sum_{i=1}^n \sum_{j=1}^m \epsilon_i \phi(z_i, z_j, \hat{\theta}) \qquad (10)$$
$$= \frac{1}{m} \sum_{i=1}^n \epsilon_i \phi_i(\hat{\theta})$$

Specifically, suppose when $\pi_i = 1$ for $\forall i = 1, 2, ..., n$, the subset is the same as the full set. In this situation, the $\hat{\theta}_\epsilon = \hat{\theta}$ such that $l_j(\hat{\theta}_\epsilon) - l_j(\hat{\theta}) = 0$ for $\forall j = 1, 2.., m$. Based on this analysis, we have the Lemma 1. For clear notion, we use bold letters to represent random variables, such that the $\phi_i(\hat{\theta})$ and ϵ_i are the realization of the random variables ϕ and ϵ, respectively.

Lemma 1. *The expectation of the influence function $\phi_i(\hat{\theta})$ over training distribution P is always 0, which means:*

$$\mathbb{E}_P(\phi) \approx \frac{1}{n} \sum_{i=1}^n \phi_i(\hat{\theta}) = 0 \qquad (11)$$

According to the Eq. (10), minimizing test risk is equivalent to minimizing the objective function $\frac{1}{n} \sum_{i=1}^n \epsilon_i \phi_i(\hat{\theta})$. Actually, this objective function is empirical form of the $\mathbb{E}_P(\phi \times \epsilon)$ from which we derive the Lemma 2.

Lemma 2. *The subset-model $\hat{\theta}_\epsilon$ performs not worse than the full-set-model $\hat{\theta}$ in terms of test risk $\mathcal{R}(Q')$ if ϵ and ϕ are negative correlated:*

$$Cov(\phi, \epsilon) \leq 0 \qquad (12)$$

The Lemma 2 gives instruction that making the perturbation ϵ negative correlated with IF ϕ can ensure better subset-model $\hat{\theta}_\epsilon$. To this end we can let the $\epsilon(\phi)$ a decreasing function. The proof of Lemma 1 and Lemma 2 can be seen in the supplementary appendix A and B, respectively.

---

[2]We assume that all elements in $\vec{\epsilon}$ are small, and each training sample z_i influences the test risk independently, hence we add up all terms linearly for simplicity of implementation.

Deterministic v.s. Probabilistic sampling

Similar to the Eq.(3), the expectation of ϕ on subset via the observation variable O can be acquired by:

$$\mathbb{E}_O(\sum_{i,O_i=1}^n \phi_i(\hat{\theta})) = \sum_{i=1}^n \mathbb{E}_{O_i}(O_i \times \phi_i(\hat{\theta})) = \sum_{i=1}^n \pi_i \phi_i(\hat{\theta}) \tag{13}$$

However, the objective function in Eq. (10) is defined on perturbation ϵ instead of sampling probability π, so that we need to bridge the gap between them:

$$\{\epsilon_i^\star\}_{i=1}^n = \arg\min_{\vec{\epsilon}} f(\vec{\epsilon}) = \arg\min_{\vec{\epsilon}} n(f(\vec{\epsilon}) + \frac{1}{n}\sum_{i=1}^n \phi_i(\hat{\theta}))$$

$$= \arg\min_{\vec{\epsilon}} \sum_{i=1}^n (n\epsilon_i + 1)\phi_i(\hat{\theta}) \tag{14}$$

The Eq. (14) holds because from Lemma 1 we know the $\mathbb{E}_P(\phi) = 0$. Here we assume that $\epsilon \in [-\frac{1}{n}, 0]$ [3], therefore, if we let $\pi = n\epsilon + 1 \in [0, 1]$, then the perturbation is transformed to sampling probability, because the Eq.(13) and Eq. (14) are in the same form.

In fact, the Eq.(13) has closed form of optimal π_i:

$$\pi_i^\star = \begin{cases} 1 & \phi_i \leq 0 \\ 0 & \phi_i > 0 \end{cases} \tag{15}$$

This form of sampling is termed as *Data dropout* in (Wang, Huan, and Li 2018) while here we call it *Deterministic sampling*, because it simply sets a threshold and selects samples deterministically. By contrast, sampling with a continuous function $\pi(\phi) \in [0, 1]$ is called *Probabilistic sampling* since each sample has a probability to be selected. Most of sampling studies belong to probabilistic sampling methods: (Wang, Zhu, and Ma 2018) builds π^{mMSE} and π^{mVc} based on *A-optimality* and *L-optimality* respectively, and (Ting and Brochu 2018) uses $\pi_i \propto \|\psi_\theta(z_i)\|$.

Analysis of sampling functions

For Dropout method, the sample's influence over a Q' is the essential criterion for sampling, such that the obtained subset-model ends up being optimal only for a Q'. However, the subset-model's robustness to distribution shift (Hu et al. 2018) is also a concern. That is, for a set of distributions around the empirical one, whether our subset-model can still maintain its performance. In this work, we postulate that the Influence-based subsampling confronts to trade the subset-model's performance on a specific Q' off its distributional robustness. In this viewpoint, the Dropout method is *overly confident* on a Q' at the expense of deteriorating generalization ability, hence it is reasonable to measure and control this *confidence degree* for our subsampling methods.

Considering an uncertainty set $\mathcal{Q} = \{Q \mid Q \ll P, D_{\chi^2}(Q\|P) \leq \delta, \delta \geq 0\}$, where the $Q \ll P$ denotes that

---

[3]The $\epsilon = 0$ means no perturbation is applied, while the $\epsilon = -\frac{1}{n}$ means a sample is totally dropped in objective function, here all perturbations are assumed within this interval.

Q is absolutely continuous w.r.t. P, and $D_{\chi^2}(\cdot\|\cdot)$ means χ^2-divergence:

$$D_{\chi^2}(Q\|P) = \mathbb{E}_P[\frac{1}{2}(1 - \frac{dQ}{dP})^2] \tag{16}$$

The $\mathcal{Q}$ is a χ^2-divergence distribution ball, indicating all neighborhoods of the empirical distribution P. The worst-case risk $\mathcal{R}_{\hat{\theta}_\epsilon}(Q)$ is defined as the supremum of the risk over any $Q \in \mathcal{Q}$ (Bagnell 2005):

$$\mathcal{R}_{\hat{\theta}_\epsilon}(Q) \triangleq \sup_{Q \in \mathcal{Q}} \{\mathbb{E}_Q[l(\hat{\theta}_\epsilon; Z)]\} \tag{17}$$

From the (Duchi and Namkoong 2018), the dual form of the Eq. (17) is:

$$\mathcal{R}_{\hat{\theta}_\epsilon}(\eta) = \inf_{\eta \in \mathbb{R}} \left\{ \sqrt{2\delta + 1} \times \mathbb{E}_P[(l(\hat{\theta}_\epsilon; Z) - \eta)_+^2]^{\frac{1}{2}} + \eta \right\} \tag{18}$$

where the η is the dual variable. This duality transforms the supremum in Eq.(17) to a convex function on the empirical distribution P, thus allowing us to measure the worst-case risk quantitatively. Before focusing on analyzing how the worst-case risk $\mathcal{R}_{\hat{\theta}_\epsilon}(Q)$ changes with the different sampling functions in Theorem 3, we need to introduce two terms:

Definition 1. *A function f(x): $\mathbb{R}^d \to \mathbb{R}$ is said to be Lipschitz continuous with constant ξ if $\|\nabla f(x) - \nabla f(y)\| \leq \xi\|x - y\|, \forall x, y \in \mathbb{R}^d$.*

Definition 2. *A function f(x) has σ-bounded gradients if $\|\nabla f(x)\| \leq \sigma$ for all $x \in \mathbb{R}^d$.*

Theorem 3. *Let η^* the optimal dual variable η that achieves the infimum in the Eq. (18), and the perturbation function $\epsilon(\phi)$ has σ-bounded gradients. Then, the worst-case risk $\mathcal{R}_{\hat{\theta}_\epsilon}(\eta^*)$ is a Lipschitz continuos function w.r.t. the IF vector $\vec{\phi} = (\phi_1, \phi_2, .., \phi_n)^\top$ where we have the Lipschitz constant $\xi = \mathcal{O}(\sigma\frac{\sqrt{2\delta+1}}{n})$, that is*

$$\|\nabla_{\vec{\phi}}\mathcal{R}_{\hat{\theta}_\epsilon}(\eta^*)\| \leq \sigma\frac{\sqrt{2\delta + 1}}{n} \times \sqrt{\sum_{i=1}^n \phi_i^2}$$

The Theorem 3 relate the change rate of the worst-case risk to the gradient bound σ of the perturbation function $\epsilon(\phi)$. For the Dropout method, its sampling function Eq.(15) has unbounded gradient since it is inconsistent at the zero point, causing the $\sigma \to \infty$. This property makes $\mathcal{R}_{\hat{\theta}_\epsilon}(\eta^*)$ no longer Lipschitz continuous and suffers sharp fluctuation. By contrast, our probabilistic methods can adjust the confidence degree by tuning the σ. This is crucial to avoid over confidence on a specific Q' that leads to large risk on other $Q \in \mathcal{Q}$. In fact, our experiments bear out that our probabilistic methods maintain its performance out-of-sample with proper σ, while the Dropout method often crashes. The proof of Theorem 3 can be found in Appendix C.

Surrogate metric for confidence degree

Nevertheless, we find that σ is the determinant of confidence degree, it is still intractable to measure this degree quantitatively, which is important to guide our methods use in

practice. Empirically, to deal with over fitting, practitioners prefer adding constraints on the model's parameters θ, e.g. l_2-norm regularizer. In our theory, we propose to apply the $\Gamma(\vec{\phi}) = \|\hat{\theta}_\epsilon - \hat{\theta}\|^2$ to evaluate the confidence degree over a specific Q'. We term the $\Gamma(\vec{\phi})$ a *surrogate metric* for confidence degree, and prove in Theorem 4 that it is reasonable because the $\Gamma(\vec{\phi})$ has the same magnitude of Lipschitz constant as the $\mathcal{R}_{\hat{\theta}_\epsilon}(\eta^*)$. In detail, the worst-case risk and our surrogate metric share the same change rate corresponding to the sampling function's gradient bound σ.

Theorem 4. *Let the perturbation function $\epsilon(\phi)$ has σ-bounded gradient, and the $|\epsilon(\phi)|$ is bounded by $\tau \in \mathbb{R}^+$, that is $|\epsilon(\phi)| \leq \tau$. We have the parameter shift $\Gamma(\vec{\phi}) = \|\hat{\theta}_\epsilon - \hat{\theta}\|^2$ is Lischitz continuous with its Lipschitz constant $\xi = \mathcal{O}(\sigma\tau)$. Specifically for $\tau = \frac{1}{n}$, we have $\xi = \mathcal{O}(\frac{\sigma}{n})$.*

The Theorem 4 is helpful to measure our sampling methods confidence degree in practice if the radius δ of the χ^2-divergence ball is unknown. Theoretically, a relatively small σ guarantees more robust model. The proof of the Theorem 4 can be found in the supplementary appendix D.
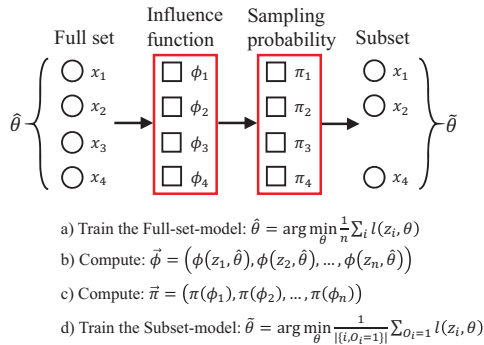
a) Train the Full-set-model: $\hat{\theta} = \arg\min_\theta \frac{1}{n} \Sigma_i l(z_i, \theta)$

b) Compute: $\vec{\phi} = \left(\phi(z_1, \hat{\theta}), \phi(z_2, \hat{\theta}), ..., \phi(z_n, \hat{\theta})\right)$

c) Compute: $\vec{\pi} = \left(\pi(\phi_1), \pi(\phi_2), ..., \pi(\phi_n)\right)$

d) Train the Subset-model: $\tilde{\theta} = \arg\min_\theta \frac{1}{|\{i, O_i = 1\}|} \Sigma_{O_i = 1} l(z_i, \theta)$

Figure 2: Our unweighted subsampling framework.

Implementation

In this section, the unweighted subsampling method is incorprated in our framework, shown in Fig.2: we train $\hat{\theta}$ with the full set data, and calculate the IF vector $\vec{\phi} = (\phi_1, \phi_2, ..., \phi_n)$ on training set with the $\hat{\theta}$. The sampling probabilities are acquired with the designed probabilistic sampling function $\pi(\cdot)$ afterwards. We will discuss the two basic modules of this framework: 1) calculating the IF and 2) designing probabilistic sampling functions.

Calculating influence functions

The IF in Eq.(7) can be calculated in two steps: first calculating the *inverse Hessian-vector-product* (HVP) $\nabla_\theta l_j(\hat{\theta})^\top H_{\hat{\theta}}^{-1}$, then multiply it with the $\nabla_\theta l_i(\hat{\theta})$ for each training sample. To handle the sparse scenarios when the $H_{\hat{\theta}}$ has high dimensions, (Martens 2010) proposes to transform the inverse HVP into an optimization problem: $H_{\hat{\theta}}^{-1} v \equiv \arg\min_t \{\frac{1}{2} t^\top H_{\hat{\theta}} t - v^\top t\}$, and solve it with Newton conjugate gradient (Newton-CG) method. Moreover, the (Agar-
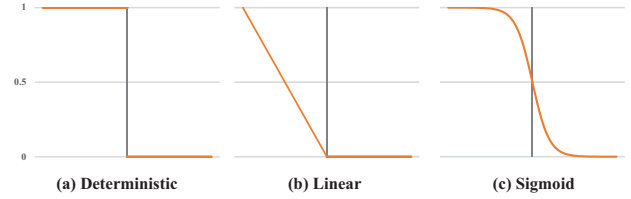
Figure 3: A family of sampling functions, the x axis is value of influence function ϕ and the y axis is the probability π.

wal, Bullins, and Hazan 2017) proves that the stochastic estimation makes the calculation feasible when the loss function $l(\cdot)$ is non-convex in θ. These works ensure our framework's feasibility in both convex and non-convex scenarios. Without loss of generality, we mainly focus on convex scenarios.

When the CG converges slowly because of ill-conditioned sub-problem, the mixed preconditioner $\bar{M}$ is useful to reduce CG steps(Nash 1985; Hsia, Chiang, and Lin 2018):

$$\bar{M} = \alpha \times diag(H_\theta) + (1 - \alpha) \times I \qquad (19)$$

where the α is a weight parameter, I is the identical matrix and H_θ is the Hessian matrix. Specifically for logistic regression model, its diagonal elements on H_θ is:

$$(H_\theta)_{kk} = 1 + C \sum_{i=1}^n (\hat{y}_i - y_i) x_{ik}^2 \qquad (20)$$

where the C is the regularization parameter. Our experiments demonstrate that the Mixed PCG is efficacious for speeding up the calculation of IF.

Probabilistic sampling functions

From Lemma 2, better subset-model is ensured with a decreasing function $\pi(\phi)$. Furthermore, the Theorem 3 and 4 prove that the gradient bound σ of $\pi(\phi)$ can be adjusted for confidence degree over a Q'. We can design a family of probabilistic sampling functions with tunable hyper parameter w.r.t. σ. Here we develop two basic functions, termed as *Linear sampling* and *Sigmoid sampling*.

Linear sampling. Inspired by (Ting and Brochu 2018), which builds $\pi_i \propto \|\psi_\theta(z_i)\|$ as $\pi_i = \max\{\alpha, \min\{1, \lambda\|\psi_\theta(z_i)\|\}\}$, we design a Linear sampling function where we let $\pi_i \propto -\phi_i(\hat{\theta})$:

$$\pi_i = n\epsilon(\phi_i) + 1 = \max\{0, \min\{1, -\alpha\phi_i\}\} \qquad (21)$$

where $\alpha \in \mathbb{R}^+$. It is easy to prove the gradient bound σ of $\pi(\phi)$ is α, thus the degree of confidence relies on the α for the Linear sampling. It is a little different from the $\pi_i \propto \|\psi_\theta(z_i)\|$, because the ϕ_i can be both negative and positive, which means many samples can have zero probability to be sampled. If we set a relatively high sampling ratio, like 90% or higher, we will never get enough samples in our subset. Empirically, we find randomly picking up the negative π samples reaches relatively good results.
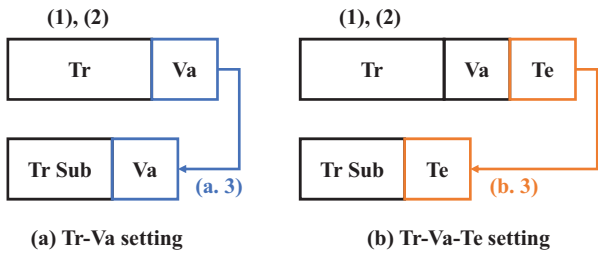
(a) Tr-Va setting   **(b) Tr-Va-Te setting**

Figure 4: Tr-Va v.s. Tr-Va-Te setting.

(a) Breast cancer   (b) MNIST

Figure 5: Test ACC with noisy labels (40% being flipped).

Sigmoid sampling. The Sigmoid function is generally used in logistic regression to scale the outputs into $[0, 1]$, which indicates probability of each class, such that here we can use it to transform ϕ_i to probability as following:

$$\pi_i = n\epsilon(\phi_i) + 1 = \frac{1}{1 + e^{\frac{\alpha\phi_i}{\max(\{\phi_i\}) - \min(\{\phi_i\})}}} \qquad (22)$$

where $\alpha \in \mathbb{R}^+$. For the Sigmoid function, we can still adjust the α to make the probability distribution more flat or steep, thereby control the confidence degree.

Experiments

In this section, we present data sets and experiment settings at first, and introduce several baselines for comparison. After that, we do experiments to evaluate our methods in terms of effectiveness, robustness and efficiency.

Data sets

We perform extensive experiments on various public data sets which conclude many domains, including computer vision, natural language processing, click-through rate prediction, etc. Additionally, we test the methods on the **Company** data set which contains around **100 million** samples with over **10 million** features. They are queries collected from a real world recommender system, whose feature set contains user history behaviour, item's side information and contextual information, such as time and location. These data sets range from small to large, from low dimensions to high dimensions, which can testify the methods effectiveness and robustness in diverse scenarios. The data set statistics and more details about preprocessing on some data sets are described in appendix E.

Considered baselines

We select the model trained on full data set and other three sampling approaches as the baselines for comparison: Optimal sampling (Ting and Brochu 2018), Data dropout (Wang, Huan, and Li 2018) and Random sampling.

A. **Optimal sampling.** It is a recent research in weighted subsampling that uses sampling probability proportional to $\|\psi_\theta\|$: $\pi_i = \max\{\alpha, \min\{1, \lambda\|\psi_\theta(z_i)\|\}\}$. This method aims at best approaching full set performance such that it cannot overtake full set theoretically.

B. **Data dropout.** It is an unweighted subsampling approach which adopts a simple sampling strategy that is dropping out unfavorable samples whose $\phi_i > 0$.
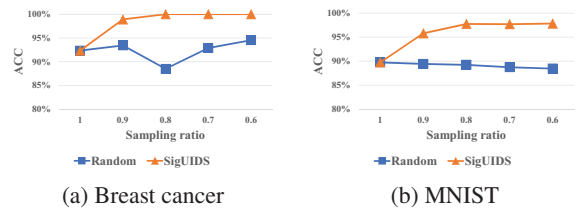
C. **Random sampling.** It is simple random selection on Tr which means all samples share same probability. Theoretically, this strategy cannot win over full set as well.

Experiment protocols

In our experiments, we use a Tr-Va-Te setting which is different from the Tr-Va setting as many previous work do (see the Fig. 4). Both settings proceed in three steps, and share the same first two steps: 1) training model $\hat{\theta}$ on the full Tr, predicting on the Va, then computing the IF; 2) getting sampling probability from the IF, doing sampling on Tr to get the subset, then acquiring the subset-model $\tilde{\theta}$. In the third step, we introduce an additional out-of-sample test set (Te) to test the $\tilde{\theta}$ (step (b.3)) rather than testing the $\tilde{\theta}$ on the Va (step (a.3)). The reason is that if we use the $\hat{\theta}$'s validation loss on the Va to guide the subsampling and then train the subset-model $\tilde{\theta}$, the testing result of $\tilde{\theta}$ on the Va cannot convince us of its generalization ability.

In fact, our framework is applicable for both convex and non-convex models, and we mainly focus on subsampling theory in this work. For implementation simplicity, we use logistic regression in all experiments. Besides, to ensure that our methods indeed achieve good performance in terms of the metrics they optimized for, i.e., we use the logistic loss for computing the influence function, such that the logloss is used as the metric in all experiments. More details about experimental settings can be found in appendix F.
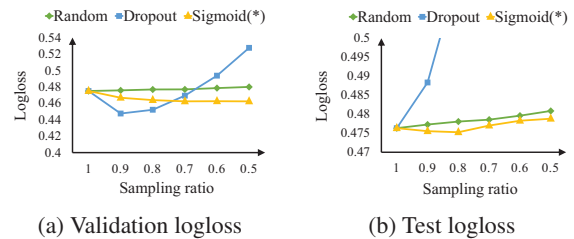

(a) Validation logloss   (b) Test logloss

Figure 6: Comparison on logloss between Va set and Te set.

Experiment observations

Result 1: Effectiveness. The experimental results are shown in the Table 2 where sampling ratio is set as 95% for all sampling methods. The average test logloss [4] values are listed in each column. The bold letters indicate logloss

---

[4] repeat 10 times sampling from the same Tr set

Table 2: Average logloss evaluated on the out-of-sample Te set when sampling ratio is set to 95%.

	Full set	Random	OptLR	Dropout	Lin-UIDS[*]	Sig-UIDS[*]
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__**
Company	0.1955	0.1956	0.1958	0.1964	**__0.1952__**	**0.1953**

[*] The UIDS is the abbreviation of our **U**nweighted **I**nfluence **D**ata **S**ubsampling methods. The Lin- and Sig-indicates the incorporated Linear and Sigmoid sampling functions, respectively.
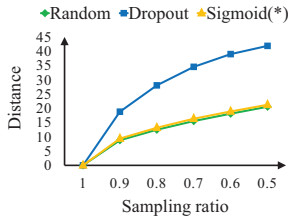

Figure 7: The confidence degree $\|\tilde{\theta} - \hat{\theta}\|_2^2$.

Table 3: Time costs of computing the influence function on whole training set.

Dataset	# samples	# features	Time cost (sec)
diabetes[a]	768	8	0.03
news20[a]	19,954	1,355,192	2.77
cifar10[a]	60,000	3,072	5.85
criteo1%[a]	456,674	1,000,000	38.85
Avazu-app[b]	14,596,137	1,000,000	166.56
Avazu-site[b]	25,832,830	1,000,000	310.66

[a] Run on the Intel i7-6600U CPU @2.60GHz.
[b] Run on the Intel Xeon CPU E5-2670 v3 @2.30GHz.

less than the full-set-model, and the underlying ones are the best across the row. It can be seen that our Sig- and Lin-UIDS overhaul the full-set-model on most of data sets, while Dropout often fails. Besides, due to the high variance incurred by weight term, the OptLR method severely suffers. In a nutshell, the Sig-UIDS performs the best on 5 of 14 datasets, and both Lin-UIDS and Dropout achieves 4. That means over-confidence sometimes is beneficial on those homogeneous data sets, e.g. the MNIST, but the Dropout fails on all relatively large-scale and heterogeneous data sets. The probabilistic sampling methods have universal superiority over others, since it keeps robustness on a set of distributions $\mathcal{Q}$, rather than a specific $\mathcal{Q}'$ (the Va set).

Our unweighted method can downweight the bad cases which cause high test loss to the our model, which is an important reason of its ability to improve result with less data. To show the performance of our methods in noisy label situation, we perform addtional experiments with some training labels being flipped. The results show the enlarging superiority of our subsampling methods in Fig. 5.

Result 2: Robustness. In Fig. 6, we can see that the Dropout method performs very well on Va set, however, it fails in out-of-sample test. To illustrate how the proposed surrogate metric for confidence degree works, we set sampling ratio from large to small, then observe how the surro-

gate metric $\Gamma(\vec{\phi})$ changes. See in Fig. 7, the Dropout causes large shift, while our Sig-UIDS has as small shift as the Random sampling. This phenomenon coincides with our Theorem 4 that the Lipschitz constant $\xi = \mathcal{O}(\frac{\sigma}{n})$. With the presence of proper σ in Sig-UIDS, the majority of π are around 0.5, which makes the sampling process more smooth.

Result 3: Efficiency. The Table 3 shows summary of running time. For most of the data sets, our method can calculate the IF within one minute. With large and sparse data sets, our method can achieve computation within ten minutes, which is acceptable in practice.

Conclusion & Future Work

In this work, we theoretically study the unweighted subsampling with IF, then propose a novel unweighted subsampling framework and design a family of probabilistic sampling methods. The experiments show that 1) different from the previous weighted methods, our unweighted method can acquire the subset-model that indeed wins over the full-set-model on a given test set; 2) it is crucial to evaluate the confidence degree over the empirical distribution for enhancing

our subset-model $\tilde{\theta}$'s generalization ability.

Although our framework of the Unweighted Influence Data Subsampling (UIDS) method succeeds in improving model accuracy, there are still some interesting ideas remain to be explored. Since our framework is applicable both for convex and non-convex models, we can further testify its performance on those non-convex models, e.g. Deep Neural Networks. Another direction is to develop better approaches to deal with the over fitting issue, e.g. build a validation set selection scheme. Besides, we plan to implement our method in industry in the future.

Acknowledgement

References

Agarwal, N.; Bullins, B.; and Hazan, E. 2017. Second-order stochastic optimization for machine learning in linear time. *The Journal of Machine Learning Research* 18(1):4148–4187.

Ai, M.; Yu, J.; Zhang, H.; and Wang, H. 2018. Optimal subsampling algorithms for big data generalized linear models. *arXiv preprint arXiv:1806.06761*.

Bagnell, J. A. 2005. Robust supervised learning. In *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, 714–719.

Cook, R. D., and Weisberg, S. 1980. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics* 22(4):495–508.

Duchi, J. C., and Namkoong, H. 2018. Learning models with uniform performance via distributionally robust optimization. *ArXiv* abs/1810.08750.

Fithian, W., and Hastie, T. 2014. Local case-control sampling: Efficient subsampling in imbalanced data sets. *Annals of statistics* 42(5):1693.

Freund, Y., and Schapire, R. E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1):119–139.

Hsia, C.-Y.; Chiang, W.-L.; and Lin, C.-J. 2018. Preconditioned conjugate gradient methods in truncated newton frameworks for large-scale linear classification. In *Asian Conference on Machine Learning*, 312–326.

Hu, W.; Niu, G.; Sato, I.; and Sugiyama, M. 2016. Does distributionally robust supervised learning give robust classifiers? In *ICML*.

Hu, W.; Niu, G.; Sato, I.; and Sugiyama, M. 2018. Does distributionally robust supervised learning give robust classifiers? In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2034–2042.

Huber, P. J. 2011. *Robust statistics*. Springer.

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.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *24th Annual Conference on Neural Information Processing Systems 2010 (NIPS)*, 1189–1197.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2999–3007.

Malisiewicz, T.; Gupta, A.; and Efros, A. A. 2011. Ensemble of exemplar-svms for object detection and beyond. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, 89–96.

Martens, J. 2010. Deep learning via hessian-free optimization. In *ICML*, volume 27, 735–742.

Nash, S. G. 1985. Preconditioning of truncated-newton methods. *SIAM Journal on Scientific and Statistical Computing* 6(3):599–616.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*.

Schnabel, T.; Swaminathan, A.; Singh, A.; Chandak, N.; and Joachims, T. 2016. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352*.

Sharchilev, B.; Ustinovsky, Y.; Serdyukov, P.; and de Rijke, M. 2018. Finding influential training samples for gradient boosted decision trees. *arXiv preprint arXiv:1802.06640*.

Sliwinski, J.; Strobel, M.; and Zick, Y. 2019. Axiomatic characterization of data-driven influence measures for classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, 718–725.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Ting, D., and Brochu, E. 2018. Optimal subsampling with influence functions. In *Advances in Neural Information Processing Systems*, 3650–3659.

Wang, T.; Huan, J.; and Li, B. 2018. Data dropout: Optimizing training data for convolutional neural networks. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, 39–46. IEEE.

Wang, H.; Zhu, R.; and Ma, P. 2018. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association* 113(522):829–844.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019. Theoretically principled trade-off between robustness and accuracy. In *ICML*.

Zhang, X.; Zhu, X.; and Wright, S. J. 2018. Training set debugging using trusted items. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 4482–4489.