Data Valuation using Reinforcement Learning

Jinsung Yoon [1] **Sercan Ö. Arık** [1] **Tomas Pfister** [1]

Abstract

Quantifying the value of data is a fundamental problem in machine learning and has multiple important use cases: (1) building insights about the dataset and task, (2) domain adaptation, (3) corrupted sample discovery, and (4) robust learning. We propose Data Valuation using Reinforcement Learning (DVRL), to adaptively learn data values jointly with the predictor model. DVRL uses a data value estimator (DVE) to learn how likely each datum is used in training of the predictor model. DVE is trained using a reinforcement signal that reflects performance on the target task. We demonstrate that DVRL yields superior data value estimates compared to alternative methods across numerous datasets and application scenarios. The corrupted sample discovery performance of DVRL is close to optimal in many regimes (i.e. as if the noisy samples were known apriori), and for domain adaptation and robust learning DVRL significantly outperforms state-of-the-art by 14.6% and 10.8%, respectively.

1. Introduction

Data is an essential ingredient of machine learning – it is well known that training on larger-scale and higher-quality datasets results in superior models (Hestness et al., 2017; Najafabadi et al., 2015). However, collecting such large-scale and high-quality datasets can be challenging and costly, as one needs to determine which data samples are most useful for the target task and then label them correctly.

Recent work (Toneva et al., 2019) suggests that not all samples are equally useful to learn from, particularly so for Deep Neural Networks (DNNs). Many datasets contain low-quality (e.g. due to measuring hardware) or incorrectly-labeled samples (e.g. due to human errors), and in those scenarios similar or even higher performance may be obtained by removing a significant portion of training samples (Ferdowsi et al., 2013; Frenay & Verleysen, 2014). Moreover, datasets may contain a mismatch between the train and test sets (e.g. different location or time), and in those cases higher performance can be obtained by carefully selecting the samples most relevant for the test scenario from the training set (Ngiam et al., 2018; Zhu et al., 2019). Because of the ubiquity of these scenarios, accurately quantifying the values of training samples has a great potential for improving model performance on real-world datasets. In addition to improving model performance, assigning a value to individual datum can also enable new use cases. It can be used to suggest better practices for data collection, e.g. what kinds of additional data would benefit the most. Organizations that sell data can use it for pricing of each datum. Finally, it can be used to construct large-scale training datasets in a cheap way, e.g., by web searching using the labels as keywords and filtering away less valuable data.

So how does one evaluate the value of a single datum? At the full dataset granularity, it is straightforward: one can simply train a model on the entire dataset and use its performance on a testing set as its value. However, evaluating the value of *a single datum* is far more difficult – especially so for complex models such as DNNs on large-scale datasets as it is computationally infeasible to train them on all subsets. To tackle this, early explorations have recently been performed with permutation-based methods such as Influence Functions (Koh & Liang, 2017) and game theory-based methods such as Data Shapley (Ghorbani & Zou, 2019). However, even the best current methods are far from being computationally feasible for large datasets and complex models and their data valuation performance is limited. Concurrently, meta learning-based adaptive weight assignment approaches such as (Ren et al., 2018) have been developed. Their data value mapping is typically based on gradient descent learning or other heuristic approaches that alter the conventional predictor model training dynamics, rather than prioritizing learning from high value data samples.

To address these challenges, we propose a novel approach to data valuation based on meta learning. Unlike previous works, our method integrates data valuation into the training procedure of the predictor model. This allows the predictor model to get extra supervision from samples that are more valuable for the given task, improving both predictor and

[1]Google Cloud AI, Sunnyvale, California, USA. Correspondence to: Jinsung Yoon <jinsungyoon@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

data valuation performance. To infer the data values, we propose a data value estimator (DVE) that estimates data values and selects the most valuable samples to train the predictor. This selection operation is fundamentally non-differentiable and thus conventional gradient-descent based methods cannot be used. Instead, we propose to use Reinforcement Learning (RL) such that the supervision of DVE is based on a reward that quantifies the predictor performance on a small validation set. The reward guides the optimization of the policy towards the action of optimal data valuation, given the state, input samples. Here, we treat the predictor model learning and evaluation framework as the environment, as a novel application scenario of RL-assisted machine learning.

A well-performing data value estimator can reprioritize the use of training samples as needed and enables training of high performance predictors even in the face of low quality, noisy or out-of-domain training samples. This can revolutionize how we think about dataset construction – what kind of data to collect and label – and more generally how we train deep learning models – enabling better training with a combination of low quality and high quality datasets instead of a standard single training dataset. To demonstrate the potential of DVRL, we focus on a wide range of use cases including corrupted sample discovery, robust learning and domain adaptation in a diverse range of image, tabular and language datasets. We show that DVRL significantly outperforms all notable data value estimation methods with orders of magnitude less computational cost, and in its generic form, it also outperforms specifically-designed meta learning approaches for robust learning or domain adaptation.

2. Related Work

Data valuation quantifies the contribution of individual datum to the overall performance. A commonly-used method for data valuation is leave-one-out (LOO), which quantifies the performance difference when a specific sample is removed to assign it as that sample's data value. The computational complexity of LOO scales linearly with the number of training samples and becomes prohibitively high for large-scale datasets and complex models. In addition, there are fundamental limitations in the approximation – e.g. despite being very important, one of the two exactly equivalent samples may get a low data value with LOO because high performance can be obtained by including the other sample. The method of Influence Functions (Koh & Liang, 2017; Wang et al., 2019) approximates LOO in a computationally-efficient manner. It uses the gradient of the loss function with small perturbations to estimate the data value. It requires Hessian values that are prohibitively expensive to compute for DNNs. Approximations for Hessian are possible, although they often cause performance limitations. It also inherits the major limitations of LOO.

Data Shapley (Ghorbani & Zou, 2019) is another approach for data valuation. Shapley values are motivated by game theory (Shapley, 1953) and are commonly used in feature attribution problems such as relating predictions to input features (Lundberg & Lee, 2017). For Data Shapley, the prediction performance of all possible subsets is considered and the marginal performance improvement is used as the data value. The computational complexity for obtaining the exact Shapley value is exponential with the number of samples. Therefore, Monte Carlo sampling and gradient-based estimation are used to approximate them. However, even with these, the computational cost still remains high (indeed much higher than LOO) due to the models needing to be re-trained for each test combination. In addition, the approximations may result in fundamental limitations for data valuation performance – e.g. with Monte Carlo approximation, the ratio of tested combinations compared to all possible combinations decreases exponentially.

In addition to their approximation limitations and high computational cost, all the aforementioned methods for data valuation are decoupled from predictor model training, causing performance limitations. To address this, we design DVRL such that learning is performed jointly for the data value estimator and the corresponding predictor model, enabling the predictor model to learn how to optimize itself for high value samples, and resulting in data valuation is guided with a high-performance predictor model.

Meta learning for adaptive weight assignment has been utilized for various use cases such as robust learning, domain adaptation, and corrupted sample discovery. ChoiceNet (Choi et al., 2018) explicitly models output distributions and uses the output correlations to improve robustness. Li et al. (2019) combines meta learning with standard stochastic gradient update with generated synthetic noise for robust learning. Shen & Sanghavi (2019) alternates the processes of selecting the samples with low error and model training to improve robustness. Shu et al. (2019) uses DNNs to model the relations between current loss and the corresponding sample weights, and utilizes a meta learning framework for weight assignment. Köhler et al. (2019) estimates the uncertainty to discover the noisy labels and relabels mislabeled samples to improve the predictor model. Gold Loss Correction (Hendrycks et al., 2018) uses a clean validation set to recover the label corruption matrix to re-train the predictor model with corrected labels. Learning to Reweight (Ren et al., 2018) proposes a single gradient descent step guided with validation set performance to reweight the training batch. Domain Adaptive Transfer Learning (Ngiam et al., 2018) introduces importance weights (based on the prior label distribution match) to scale the training samples for transfer learning. MentorNet (Jiang et al., 2018) proposes curriculum learning to learn the order of mini-batches.

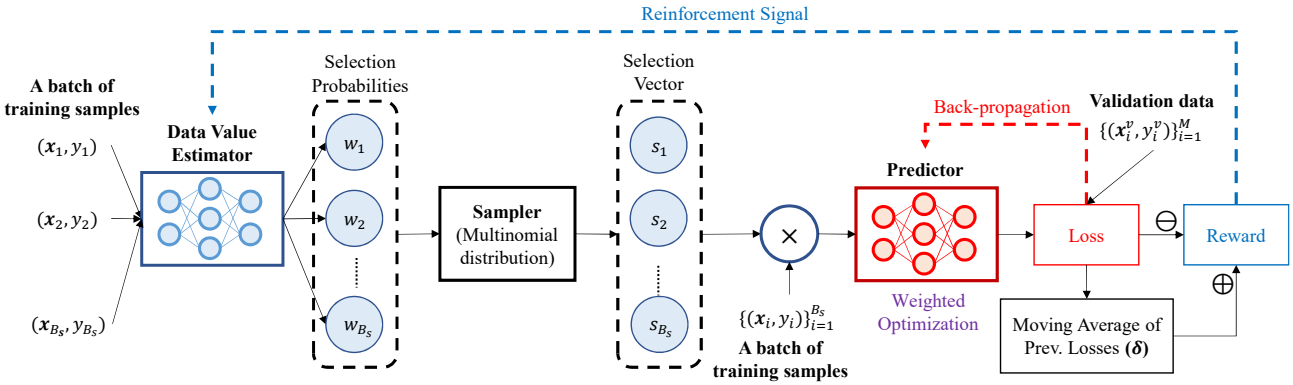

Figure 1. Block diagram of the DVRL framework for training. A batch of training samples is input to DVE (with shared parameters across the batch) that outputs selection probabilities: $w_i = h_\phi(\mathbf{x}_i, y_i)$ of a multinomial distribution. The sampler, based on this distribution, returns the selection vector $\mathbf{s} = (s_1, ..., s_{B_s})$ where $s_i \in \{0, 1\}$ and $P(s_i = 1) = w_i$. The target task predictor model is trained only using the samples with selection vector $s_i = 1$ using conventional gradient-descent optimization. The selection probabilities w_i rank the samples according to their importance – these importance scores are used as data values. The loss of the predictor model is evaluated on a small validation set and is compared to the moving average of the previous losses (δ) to determine the reward. Finally, the reinforcement signal guided by this reward updates the DVE. Block diagram at inference time is shown in supplementary materials (Section 1).

Our method, DVRL, tackles the three fundamental shortcomings of previous methods (poor approximation, computational cost, lack of joint training) by directly modeling the value of the data using a learnable DVE, for which we use RL to optimize with policy gradients. DVRL is model-agnostic and even applicable to non-differentiable target objectives. Learning is performed jointly for DVE and the corresponding predictor model to jointly improve the predictor and data valuation performance, overall yielding superior results in all of the use cases we consider.

3. Proposed Method

Framework: We denote the training dataset as $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N \sim \mathcal{P}$ where $\mathbf{x}_i \in \mathcal{X}$ is a d-dimensional feature vector, and $y_i \in \mathcal{Y}$ is a corresponding label. We consider a disjoint testing dataset $\mathcal{D}^t = \{(\mathbf{x}_j^t, y_j^t)\}_{j=1}^M \sim \mathcal{P}^t$ where the target distribution $\mathcal{P}^t$ does not need to be the same with the training distribution $\mathcal{P}$. We assume an availability of a (small) validation dataset $\mathcal{D}^v = \{(\mathbf{x}_k^v, y_k^v)\}_{k=1}^L \sim \mathcal{P}^t$ that comes from the target distribution $\mathcal{P}^t$.

DVRL consists of two learnable functions: (1) the target task predictor model f_θ, and (2) the data value estimator (DVE) model h_ϕ. The predictor model $f_\theta : \mathcal{X} \to \mathcal{Y}$ is trained to minimize a weighted loss function $\mathcal{L}_f$ on training dataset $\mathcal{D}$ (e.g. Mean Squared Error (MSE) for regression or cross entropy for classification):

$$f_\theta = \arg\min_{\hat{f} \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} h_\phi(\mathbf{x}_i, y_i) \cdot \mathcal{L}_f(\hat{f}(\mathbf{x}_i), y_i). \quad (1)$$

f_θ can be any trainable function with parameters θ, such as a DNN. The DVE model $h_\phi : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ is optimized to output weights that determine the selection likelihoods of the training samples to train the predictor model f_θ. We formulate the corresponding optimization problem as:

$$\begin{aligned} \min_{h_\phi} \quad & \mathop{\mathbb{E}}_{(\mathbf{x}^v, y^v) \sim P^t} \Big[\mathcal{L}_h(f_\theta(\mathbf{x}^v), y^v) \Big] \\ \text{s.t.} f_\theta = \quad & \arg\min_{\hat{f} \in \mathcal{F}} \mathop{\mathbb{E}}_{(\mathbf{x}, y) \sim P} \Big[h_\phi(\mathbf{x}, y) \mathcal{L}_f(\hat{f}(\mathbf{x}), y) \Big] \end{aligned} \quad (2)$$

where $h_\phi(\mathbf{x}, y)$ represents the value of the training sample $(\mathbf{x}, y)$. The DVE is a trainable function such as a DNN as assumed here. Similar to $\mathcal{L}_f$, we use MSE or cross entropy for $\mathcal{L}_h$. We use the outputs of the DVE model, $w = h_\phi(\mathbf{x}, y)$, as the *data values*. These data values can be used to rank the training data (e.g. to determine a subset of the training data) and to do sample-adaptive training (e.g. for domain adaptation).

Training: Fig. 1 and Algorithm 1 give an overview of the training procedure for DVRL. We next discuss training of each of these components.

To encourage exploration based on the uncertainty in the exponentially-large selection space, we model training sample selection in DVE stochastically. Let $w = h_\phi(\mathbf{x}, y)$ denote the probability that $(\mathbf{x}, y)$ is used to train the predictor model f_θ; $h_\phi(\mathcal{D}) = \{h_\phi(\mathbf{x}_i, y_i)\}_{i=1}^N$ is the probability distribution for inclusion of each training sample; $\mathbf{s} \in \{0, 1\}^N$ is a binary vector that represents the selected samples. If $s_i = 1/0$, $(\mathbf{x}_i, y_i)$ is selected/not selected for training the predictor model. $\pi_\phi(\mathcal{D}, \mathbf{s}) = \prod_{i=1}^N \big[h_\phi(\mathbf{x}_i, y_i)^{s_i} \cdot (1 - h_\phi(\mathbf{x}_i, y_i))^{1-s_i} \big]$ is the probability that the selection vector $\mathbf{s}$ is selected based on $h_\phi(\mathcal{D})$.

The predictor model can be trained using standard stochastic

gradient descent because it is differentiable with respect to the input. However, gradient descent-based optimization cannot be used for the DVE because the sampling process is non-differentiable. There are multiple ways to handle the non-differentiable optimization bottleneck, such as Gumbel-softmax (Jang et al., 2017) or stochastic back-propagation (Rezende et al., 2014). In this paper, we consider RL instead, which directly encourages exploration of the policy towards the optimal solution of Eq. (2). In this RL setting, action of the agent (DVE) is its data selection, and the environment, that encompasses the predictor model training and evaluation, correspondingly gives a reward for each action, based on the state of current batch of data. We adapt the REINFORCE algorithm (Williams, 1992) to optimize using the policy gradients, and obtain the rewards from a small validation set that approximates performance on the target task. Ablation studies in Section 5 demonstrate the importance of discrete selection for predictor model training and the efficacy of our proposed policy learning. As the single step reward based on the action, we use:

$$\begin{aligned}\hat{l}(\phi) &= \mathop{\mathbb{E}}_{\substack{(\mathbf{x}^v, y^v)\sim P^t, \\ \mathbf{s}\sim\pi_\phi(\mathcal{D},\cdot)}} \Big[\mathcal{L}_h(f_\theta(\mathbf{x}^v), y^v)\Big] \\ &= \int P^t(\mathbf{x}^v) \sum_{\mathbf{s}\in[0,1]^N} \pi_\phi(\mathcal{D}, \mathbf{s}) \cdot \Big[\mathcal{L}_h(f_\theta(\mathbf{x}^v), y^v)\Big] d\mathbf{x}^v,\end{aligned}$$

which has the gradient:

$$\begin{aligned}\nabla_\phi \hat{l}(\phi) &= \int P^t(\mathbf{x}^v) \sum_{\mathbf{s}\in[0,1]^N} \nabla_\phi \pi_\phi(\mathcal{D}, \mathbf{s}) \\ &\qquad \cdot \Big[\mathcal{L}_h(f_\theta(\mathbf{x}^v), y^v)\Big] d\mathbf{x}^v \\ &= \int P^t(\mathbf{x}^v) \Big[\sum_{\mathbf{s}\in[0,1]^N} \frac{\nabla_\phi \pi_\phi(\mathcal{D}, \mathbf{s})}{\pi_\phi(\mathcal{D}, \mathbf{s})} \pi_\phi(\mathcal{D}, \mathbf{s}) \\ &\qquad \cdot \Big[\mathcal{L}_h(f_\theta(\mathbf{x}^v), y^v)\Big] \Big] d\mathbf{x}^v \\ &= \int P^t(\mathbf{x}^v) \Big[\sum_{\mathbf{s}\in[0,1]^N} \nabla_\phi \log(\pi_\phi(\mathcal{D}, \mathbf{s})) \cdot \pi_\phi(\mathcal{D}, \mathbf{s}) \\ &\qquad \cdot \Big[\mathcal{L}_h(f_\theta(\mathbf{x}^v), y^v)\Big] \Big] d\mathbf{x}^v \\ &= \mathop{\mathbb{E}}_{\substack{(\mathbf{x}^v, y^v)\sim P^t, \\ \mathbf{s}\sim\pi_\phi(\mathcal{D},\cdot)}} \Big[\mathcal{L}_h(f_\theta(\mathbf{x}^v), y^v)\Big] \nabla_\phi \log(\pi_\phi(\mathcal{D}, \mathbf{s})),\end{aligned}$$

where $\nabla_\phi \log(\pi_\phi(\mathcal{D}, \mathbf{s}))$

$$= \nabla_\phi \sum_{i=1}^{N} \log \Big[h_\phi(\mathbf{x}_i, y_i)^{s_i} \cdot (1 - h_\phi(\mathbf{x}_i, y_i))^{1-s_i}\Big]$$

To improve the stability of the policy gradient-based learning, we use the moving average of the previous loss δ with a window size T as the baseline.

To provide further information to DVE, we propose to use an additional input, *marginal information*, defined as the difference between the predictions of a separate predictive model (fine-tuned or trained from scratch on the validation set) for the training samples and the original training labels respectively. *Marginal information* is defined as $m(\mathbf{x}, y) = |y - f_v(\mathbf{x})|$ where $f_v = \arg\min_{\hat{f}\in\mathcal{F}} \mathbb{E}_{(\mathbf{x}^v, y^v)\sim P^t} \Big[\mathcal{L}_f(\hat{f}(\mathbf{x}^v), y^v)\Big]$. We simply concatenate this *marginal information* to an intermediate state of the DVE network. The value of the *marginal information* increases as the level of corruption increases in the training data, and providing it to the DVE is valuable as it can decide to lower the value accordingly.

Computational complexity: DVRL models the mapping between an input and its value with a learnable function. The training time of DVRL is not directly proportional to the dataset size, but rather dominated by the required number of iterations and per-iteration complexity. One way to minimize the computational overhead is to use pre-trained models to initialize the predictor model at each iteration. Unlike alternative methods like Data Shapley, we demonstrate the scalability of DVRL to large-scale datasets such as CIFAR-100, and complex models such as ResNet-32 (He et al., 2016) and WideResNet-28-10 (Zagoruyko & Komodakis, 2016). Instead of being exponential in the data size, the training time overhead of DVRL is only twice of conventional training. Further analyses on additional computational complexity discussions and learning dynamics of DVRL are in supplementary materials (Section 2 & 5).

Algorithm 1 Pseudo-code of DVRL training

Inputs: Learning rates $\alpha, \beta > 0$, mini-batch sizes $B_p, B_s > 0$, inner iteration count $N_I > 0$, moving average window $T > 0$, training dataset $\mathcal{D}$, validation dataset $\mathcal{D}^v = \{(\mathbf{x}_k^v, y_k^v)\}_{k=1}^L$
Initialize parameters θ, ϕ, moving average $\delta = 0$
while until convergence **do**
  Sample $\mathcal{D}_B = (\mathbf{x}_j, y_j)_{j=1}^{B_s} \sim \mathcal{D}$
  for $j = 1, ..., B_s$ **do**
    Get selection probabilities: $w_j = h_\phi(\mathbf{x}_j, y_j)$
    Sample a selection vector: $s_j \sim Ber(w_j)$
  for $t = 1, ..., N_I$ **do**
    Sample $(\tilde{\mathbf{x}}_m, \tilde{y}_m, \tilde{s}_m)_{m=1}^{B_p} \sim (\mathbf{x}_j, y_j, s_j)_{j=1}^{B_s}$
    Update the predictor model:
    $\theta \leftarrow \theta - \frac{\alpha}{B_p} \sum_{m=1}^{B_p} \tilde{s}_m \cdot \nabla_\theta \mathcal{L}_f(f_\theta(\tilde{\mathbf{x}}_m), \tilde{y}_m))$
  Update the DVE model:
  $\phi \leftarrow \phi - \Big[\frac{\beta}{L} \sum_{k=1}^{L} [\mathcal{L}_h(f_\theta(\mathbf{x}_k^v), y_k^v)] - \delta\Big] \cdot \nabla_\phi \log \pi_\phi(\mathcal{D}_B, (s_1, ..., s_{B_s}))$
  Update the baseline:
  $\delta \leftarrow \frac{T-1}{T}\delta + \frac{1}{LT} \sum_{k=1}^{L} [\mathcal{L}_h(f_\theta(\mathbf{x}_k^v), y_k^v)]$

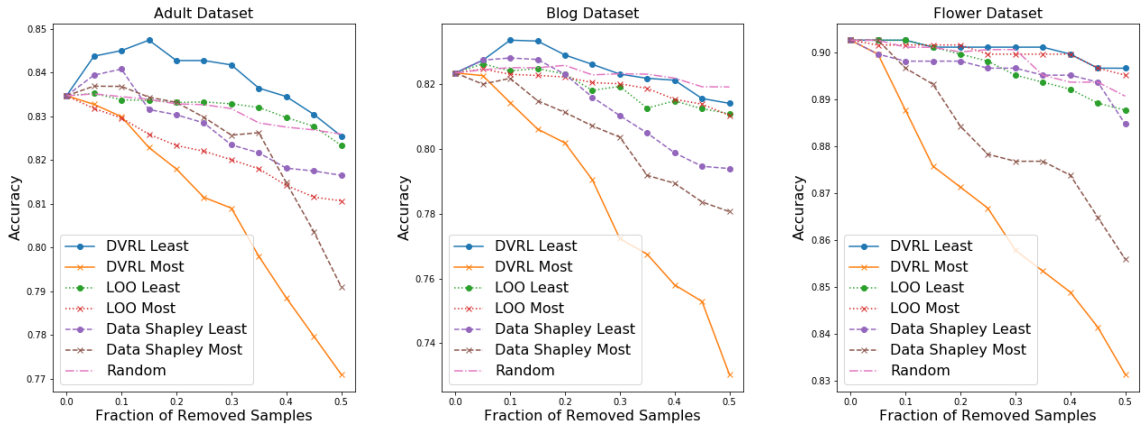

Figure 2. Performance after removing the most (marked with $\times$) and least (marked with $\bigcirc$) important samples according to the estimated data values in a conventional supervised learning setting. Similar to (Ghorbani & Zou, 2019), on Adult and Blog, we use 1,000 training and 400 validation samples; and on Flower, we use 2,000 training and 800 validation samples. We use small datasets to compare with LOO and Data Shapley which have high computational cost to train. DVRL is scalable to larger datasets, as shown in Section 4.3.

4. Experiments on Data Valuation Use Cases

We evaluate the data value estimation quality of DVRL on multiple types of datasets and use cases. Experimental details can be found in supplementary materials (Section 3) and source codes can be found in `https://github.com/google-research/google-research/tree/master/dvrl`.

Benchmark methods: We consider the following benchmarks: (1) Randomly-assigned values (Random), (2) Leave-one-out (LOO), (3) Data Shapley Value (Data Shapley) (Ghorbani & Zou, 2019). For some experiments, we also compare with (4) Learning to Reweight (Ren et al., 2018), (5) MentorNet (Jiang et al., 2018), (6) Influence Function (Koh & Liang, 2017), (7) Adversarial Discriminative Domain Adaptation (ADDA) (Tzeng et al., 2017), and (8) Domain Adversarial DNNs (DANN) (Ganin et al., 2016).

Datasets: We consider 12 public datasets (3 tabular datasets, 7 image datasets, and 2 language datasets) to evaluate DVRL in comparison to multiple benchmark methods. 3 tabular datasets are (1) Blog, (2) Adult, (3) Rossmann Store Sales; 7 image datasets are (4) HAM 10000, (5) MNIST, (6) USPS, (7) Flower, (8) Fashion-MNIST, (9) CIFAR-10, (10) CIFAR-100; 2 language datasets are (11) Email Spam, (12) SMS Spam. Details can be found in the hyper-links.

Baseline predictor models: We consider various machine learning models as the baseline predictor model to highlight the proposed *model-agnostic* data valuation framework. For Adult and Blog datasets, we use LightGBM (Ke et al., 2017), and for Rossmann Store Sales dataset, we use XGBoost and multi-layer perceptrons (MLPs) due to their superior performances on the tabular datasets. For Flower, HAM 10000, and CIFAR-10 datasets, we use Inception-v3 with top-layer fine-tuning (pre-trained on ImageNet, (Szegedy et al., 2016)) as the baseline predictor model. For Fashion-MNIST, MNIST, and USPS datasets, we use multinomial logistic regression, and for Email and SMS datasets, we use Naive Bayes model. We also use ResNet-32 (He et al., 2016) and WideResNet-28-10 (Zagoruyko & Komodakis, 2016) as the baseline models for CIFAR-10 and CIFAR-100 datasets in Section 4.3 to demonstrate the scalability of DVRL.

4.1. Removing high/low value samples

Removing low value samples from the training dataset can improve the predictor model performance, especially in the cases where the training dataset contains corrupted samples. On the other hand, removing high value samples, especially in the case of small training datasets, would decrease the performance significantly. Overall, the performance after removing high/low value samples is a strong indicator for the quality of data valuation.

We consider the conventional supervised learning setting, where all training, validation and testing datasets come from the same distribution (without sample corruption or domain mismatch). We analyze the prediction performance on the disjoint testing set after removing the high/low value samples based on the estimated data values. As shown in Fig. 2, even in the absence of sample corruption or domain mismatch, DVRL can marginally improve the prediction performance after removing some low value samples. Using only around 60%-70% of the training samples (comprised of the highest value samples), DVRL can obtain a similar perfor-

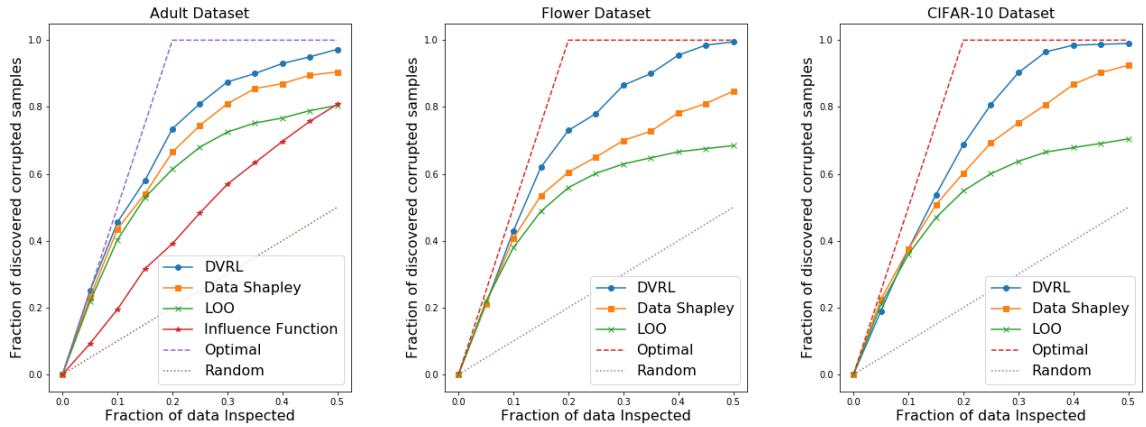

Figure 3. Discovering corrupted samples in three datasets with 20% noisy label ratio. 'Optimal' saturates at 20%, perfectly assigning the lowest data value scores to the samples with noisy labels. 'Random' does not introduce any knowledge on distinguishing clean vs. noisy labels, and thus the fraction of discovered corrupted samples is proportional to the amount of inspection. More results on Blog, Fashion-MNIST and HAM 10000 datasets can be found in supplementary materials (Section 4.3).

mance compared to training on the entire dataset. After removing a small portion (10%-20%) of the highest value samples[1], the prediction performance significantly degrades which indicates the importance of the high valued samples. Overall, DVRL shows the fastest performance degradation after removing the highest value samples and the slowest performance degradation after removing the lowest value samples in most cases, underlining the superiority of DVRL in data valuation quality compared to competing methods.

4.2. Corrupted sample discovery

Training samples may contain corrupted samples, e.g. due to cheap labeling procedures. An automated corrupted sample discovery method would be highly beneficial for distinguishing samples with clean vs. noisy labels. Data valuation can be used in this setting by having a small clean validation set to assign low data values to the potential samples with noisy labels. Ideally, all noisy labels would be assigned to the lowest data values. We consider the experimental setting of 20% noisy label ratio on 6 datasets. Fig. 3 shows that DVRL consistently outperforms all benchmarks (Data Shapley, LOO and Influence Function). DVRL can discover noisy labels almost optimally (as if we perfectly knew which samples have noisy labels), particularly for the Adult, Flower and CIFAR-10 datasets.

4.3. Robust learning with noisy labels

Ideally, noisy samples should receive low data values as DVRL converges and a high performance predictor model can be returned. We demonstrate how DVRL can reliably learn with noisy data in an end-to-end way, without removing the low-value samples as in the previous subsections.

We compare DVRL to two recently-proposed methods: MentorNet (Jiang et al., 2018) and Learning to Reweight (Ren et al., 2018) with two complex DNNs as the baseline predictor models, ResNet-32 (He et al., 2016) and WideResNet-28-10 (Zagoruyko & Komodakis, 2016), trained on CIFAR-10 and CIFAR-100 datasets.

We follow the same settings from Ren et al. (2018). For the first experiment, we use WideResNet-28-10 as the baseline predictor model and apply 40% of label noise uniformly across all classes, and with 1,000 clean samples as the validation set. For the second experiment, we use ResNet-32 as the baseline predictor model and apply 40% background noise (same-class noise to the 40% of the samples), and use 10 clean samples per class as the validation set. We test the performance on the clean testing set in both experiments. We also consider five additional benchmarks: (1) *Validation Set Only* – which only uses clean validation set for training, (2) *Baseline* – which only uses noisy training set for training, (3) *Baseline + Fine-tuning* – which is initialized with the trained baseline model on the noisy training set and fine-tuned on the clean validation set, (4) *Clean Only (60% data)* – which is trained on the clean training set after removing the training samples with noisy labels, (5) *Zero Noise* – which uses the original noise-free training set for training (100% clean training data).[2] As shown in Table 1, DVRL outperforms other robust learning methods

[1] Qualitative inspection of such small subset of the highest value samples also yields important insights about the target task.

[2] We exclude Data Shapley and LOO in this experiment due to their prohibitively-high computational cost.

Noise (predictor model)	**Uniform** (WideResNet-28-10)		**Background** (ResNet-32)	
Datasets	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Validation Set Only	46.64 ± 3.90	9.94 ± 0.82	15.90 ± 3.32	8.06 ± 0.76
Baseline	67.97 ± 0.62	50.66 ± 0.24	59.54 ± 2.16	37.82 ± 0.69
Baseline + Fine-tuning	78.66 ± 0.44	54.52 ± 0.40	82.82 ± 0.93	54.23 ± 1.75
MentorNet + Fine-tuning	78.00	59.00	-	-
Learning to Reweight	86.92 ± 0.19	61.34 ± 2.06	86.73 ± 0.48	59.30 ± 0.60
DVRL	**89.02 ± 0.27**	**66.56 ± 1.27**	**88.07 ± 0.35**	**60.77 ± 0.57**
Clean Only (60% Data)	94.08 ± 0.23	74.55 ± 0.53	90.66 ± 0.27	63.50 ± 0.33
Zero Noise	95.78 ± 0.21	78.32 ± 0.45	92.68 ± 0.22	68.12 ± 0.21

Table 1. Test accuracy for ResNet-32 and WideResNet-28-10 on CIFAR-10 and CIFAR-100 with 40% Uniform and Background noise.

Source dataset	Target dataset	Task	*Baseline*	Data Shapley	**DVRL**
Google	HAM10000	Skin Lesion Classification	.296	.378	**.448**
MNIST	USPS	Digit Recognition	.308	.391	**.472**
Email	SMS	Spam Detection	.684	.864	**.903**

Table 2. Target accuracy with domain adaptation on three scenarios: (1) using Google image search results to predict skin lesion classification on HAM 10000 data (clean), (2) using MNIST data to recognize digit on USPS dataset, (3) using Email spam data to detect spam in an SMS dataset. *Baseline* represents the predictor model which is trained with equal treatment of all training samples.

in all cases. The performance improvements with DVRL are larger with Uniform noise. Learning to Reweight loses 7.16% and 13.21% accuracy compared to the optimal case (Zero Noise); on the other hand, DVRL yields only 5.06% and 7.99% lower accuracy on CIFAR-10 and CIFAR-100 with Uniform noise. Additional results on robust learning can be found in supplementary materials (Section 4.1 & 4.2).

4.4. Domain adaptation

In some scenarios, the training dataset comes from a substantially different distribution from the validation and testing sets, and naive training methods (i.e. equal treatment of all training samples) often fail (Ganin et al., 2016; Glorot et al., 2011). We show how high-performing data valuation model can be beneficial by selecting the training samples that best match the distribution of the validation dataset.

We initially focus on the three scenarios from Ghorbani & Zou (2019), following the exactly same experimental settings. Table 2 shows that DVRL significantly outperforms *Baseline* and Data Shapley in all three scenarios. While Data Shapley needs a two step processes to construct the predictor model in domain adaptation setting, DVRL jointly optimizes the DVE and corresponding predictor model, resulting in the superior overall performance.

Next, we focus on a real-world problem where the domain differences are significant. We consider the sales forecasting problem on Rossmann dataset, which consists of sales data from four different store types. Simple data analysis (see supplementary materials (Section 7)) shows a significant discrepancy between the input feature distributions across different store types, exposing the large domain mismatch. We consider three different settings: (1) training on all store types (*Train on All*), (2) training on store types excluding the store type of interest (*Train on Rest*), and (3) training only on the store type of interest (*Train on Specific*). In all cases, we evaluate the performance on each store type separately.[3] *Train on Rest* is expected to yield the largest domain mismatch between training and testing sets, and *Train on Specific* yield the minimal (no mismatch). We evaluate the performance of *Baseline* (training without data valuation) and DVRL in 3 different settings with 2 different predictor models, XGBoost (Chen & Guestrin, 2016) and DNN (a 3-layer MLP).

As shown in Table 3, DVRL improves the performance in all settings. The improvements are most significant in *Train on Rest* setting due to the largest domain mismatch. For instance, DVRL reduces the error more than 50% for store type B predictions with XGBoost compared to *Baseline*. In *Train on All* setting, the performance improvement is still significant, showing that DVRL can distinguish the samples

[3]For example, to evaluate the performance on store type D, *Train on All* setting uses all four store type datasets for training, *Train on Rest* setting uses store types A, B and C for training, and *Train on Specific* setting only uses the store type D for training.

Predictor Model	Store Type	*Train on All*		*Train on Rest*		*Train on Specific*	
		Baseline	DVRL	*Baseline*	DVRL	*Baseline*	DVRL
XGBoost	A	0.1736	**0.1594**	0.2369	**0.2109**	0.1454	**0.1430**
	B	0.1996	**0.1422**	0.7716	**0.3607**	0.0880	**0.0824**
	C	0.1839	**0.1502**	0.2083	**0.1551**	0.1186	**0.1170**
	D	0.1504	**0.1441**	0.1922	**0.1535**	0.1349	**0.1221**
DNN	A	0.1531	**0.1428**	0.3124	**0.2014**	0.1181	**0.1066**
	B	0.1529	**0.0979**	0.8072	**0.5461**	0.0683	**0.0682**
	C	0.1620	**0.1437**	0.2153	**0.1804**	0.0682	**0.0677**
	D	0.1459	**0.1295**	0.2625	**0.1624**	0.0759	**0.0708**

Table 3. Root Mean Squared Percentage Error (RMSPE) of *Baseline* and DVRL in 3 different settings with 2 different predictor models (XGBoost, DNN) on Rossmann Store Sales dataset. We use 79% of the data as training, 1% as validation, and 20% as testing.

Ablation cases	Blog	HAM 10000	CIFAR-10
DVRL	**47.3%**	**60.2%**	**68.1%**
DVRL without the sampler, using continuous data values	44.9%	58.3%	63.7%
DVRL without a baseline in reward computation	45.8%	56.6%	62.9%
DVRL without *marginal information*	43.7%	57.1%	64.4%
Directly using *marginal information* as data values	43.1%	55.9%	62.3%

Table 4. The fraction of discovered corrupted samples after inspecting 20% of the samples with multiple variants of DVRL, on three datasets with 20% noisy label ratio (same setting with Section 4.2).

from the target distribution and often prioritizes selection of the samples from the target store type (see supplementary materials (Section 8)). In *Train on Specific* setting, even without domain mismatch, DVRL can marginally improve the performance by accurately prioritizing the important samples within the same store type, aligned with results from Fig. 2 in the conventional supervised learning setting.

Lastly, we compare DVRL to two notable domain adaptation benchmarks: Adversarial Discriminative Domain Adaptation (ADDA) (Tzeng et al., 2017) and Domain Adversarial DNNs (DANN) (Ganin et al., 2016) with DNNs as the predictor on *Train on All* setting. Table 5 shows that DVRL yields superior performance compared to ADDA and DANN, that are specifically designed for domain adaptation.

Store type	Baseline	DVRL	ADDA	DANN
A	0.1531	**0.1428**	0.1465	0.1491
B	0.1529	**0.0979**	0.1193	0.1201
C	0.1620	**0.1437**	0.1503	0.1589
D	0.1459	**0.1295**	0.1351	0.1388

Table 5. RMSPE of *Baseline*, DVRL, ADDA, and DANN in *Train on All* setting with DNNs on the Rossmann Store Sales dataset.

Discussions: Overall, the experiments (e.g., Sections 4.2, 4.3, and 4.4) suggest that DVRL brings the biggest benefits when the training dataset contains highly noisy, low-quality, or mostly out-of-distribution samples, while the validation dataset is small but clean, high-quality and in-distribution. If the training dataset is also clean, high-quality and in-distribution, the benefit of DVRL gets smaller (Section 4.1).

5. Ablation studies

In this section, we analyze the contributions of major components of DVRL. Table 4 compares the corrupted sample discovery results under various ablation cases.

Discrete representation: A straightforward idea is to use the raw outputs of DVE to scale the contributions of each sample in the loss term, without the sampler. Yet, we show the benefit of the discrete representation of DVE for data selection. The sampler encourages exploration of an extremely large action space in a systematic way, helping to converge to a better optimal solution.

Baseline for reinforcement learning: The baseline stabilizes convergence of reinforcement learning; thus, improves performance, especially on complex models and datasets.

Marginal information: The *marginal information* has valuable information as f_v (the trained model on the validation set) achieves high performance (since it is trained with small-scale high quality data). We observe that often a larger DVE model (with more iterations) is needed without *marginal information*. Yet, the overall benefit of the *marginal in-*

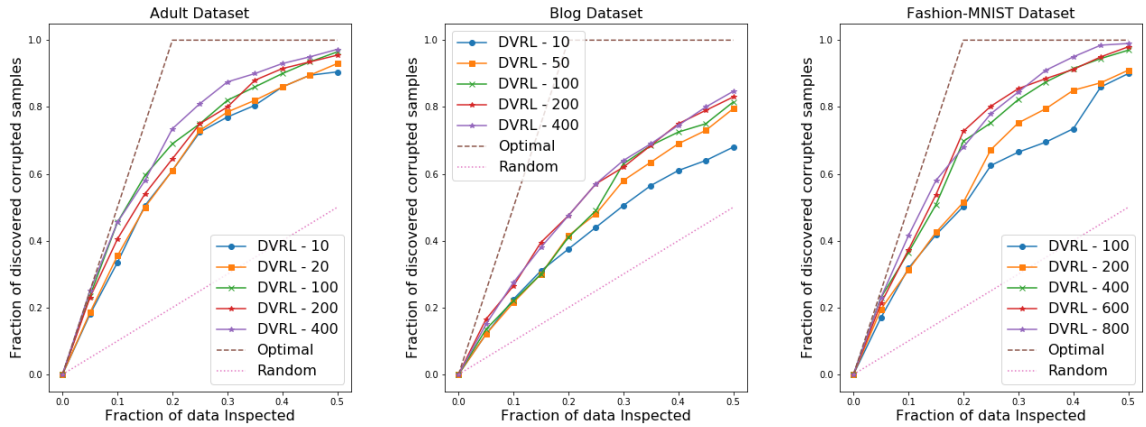

Figure 4. Analyses on the number of validation samples needed for DVRL training. We analyze the impact of the size of the validation dataset on DVRL with 3 different datasets: Adult, Blog, and Fashion MNIST for the use case of corrupted sample discovery. Similar to Section 4.2, we add 20% noise to the training samples and use DVRL to find the corrupted samples. On Adult and Fashion-MNIST datasets, DVRL needs 13% and 14% of inspected samples to identify 50% of the corrupted samples respectively - merely 3% and 4% more than the optimal cases.

formation is not as high as other components, and without DVRL, the *marginal information* itself as data value assignment seems insufficient. Note that we propose to use the output of the validation model as an additional input to the data valuation framework; thus, this can also be regarded as another contribution of our work. We observe that it is mostly helpful for the noisy sample discovery use case but not that significant in other cases such as performance improvement by low value data removal without noise or domain adaptation.

Impact of the validation dataset size: DVRL requires a validation dataset from the target distribution that the testing dataset comes from. Depending on the task, the requirements for the validation dataset may involve being noise-free in labels, being from the same domain, and/or being high quality. Acquiring such a dataset can be costly in some scenarios and it is desirable to minimize its size requirements. As shown in Fig. 4, DVRL achieves reasonable performance with around 100 to 400 validation samples. In the Adult dataset, even 10 validation samples are sufficient to achieve a reasonable data valuation quality. All these settings are often realistic in real-world scenarios.

6. Conclusions

In this paper, we propose a novel meta learning framework for data valuation which determines how likely each training sample will be used in training of the predictor model. Unlike previous work, our method integrates data valuation into the training procedure of the predictor model, allowing the predictor and DVE to improve each other's performance. We model this data value estimation task using a DNN trained using RL with a reward obtained from a small validation set that represents the target task performance. In a computationally-efficient way, DVRL can provide high quality ranking of training data that is useful for domain adaptation, corrupted sample discovery and robust learning. We show that DVRL significantly outperforms alternative methods on diverse types of tasks and datasets.

Acknowledgements

The authors would like to thank the reviewers for their insightful comments. Discussions with Kihyuk Sohn, Amirata Ghorbani, Wei Wei and Zizhao Zhang are gratefully acknowledged.

References

Chen, T. and Guestrin, C. XGBoost: A scalable tree boosting system. In *KDD*, 2016.

Choi, S., Hong, S., and Lim, S. Choicenet: Robust learning by revealing output correlations. *arXiv preprint arXiv:1805.06431*, 2018.

Ferdowsi, H., Jagannathan, S., and Zawodniok, M. An online outlier identification and removal scheme for improving fault detection performance. *IEEE Trans Neural Networks and Learning Systems*, 25(5):908–919, 2013.

Frenay, B. and Verleysen, M. Classification in the presence of label noise: A survey. *IEEE Trans on Neural Networks and Learning Systems*, 25(5):845–869, 2014.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, 2016.

Ghorbani, A. and Zou, J. Data shapley: Equitable valuation of data for machine learning. In *ICML*, 2019.

Glorot, X., Bordes, A., and Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, 2011.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In *NIPS*, 2018.

Hestness, J., Narang, S., Ardalani, N., Diamos, G. F., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically. *arXiv:1712.00409*, 2017.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*, 2017.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *ICML*, 2017.

Köhler, J. M., Autenrieth, M., and Beluch, W. H. Uncertainty based detection and relabeling of noisy image labels. In *CVPR Workshops*, 2019.

Li, J., Wong, Y., Zhao, Q., and Kankanhalli, M. S. Learning to learn from noisy labeled data. In *CVPR*, 2019.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *NIPS*, 2017.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., and Muharemagic, E. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

Ngiam, J., Peng, D., Vasudevan, V., Kornblith, S., Le, Q. V., and Pang, R. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

Shapley, L. S. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

Shen, Y. and Sanghavi, S. Learning with bad training data via iterative trimmed loss minimization. In *ICML*, 2019.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. *arXiv preprint arXiv:1902.07379*, 2019.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Toneva, M., Sordoni, A., des Combes, R. T., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. In *ICLR*, 2019.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

Wang, Z., Zhu, H., Dong, Z., He, X., and Huang, S.-L. Less is better: Unweighted data subsampling via influence function. *arXiv preprint arXiv:1912.01321*, 2019.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhu, L., Arik, S. O., Yang, Y., and Pfister, T. Learning to Transfer Learn. *arXiv preprint arXiv:1908.11406*, 2019.