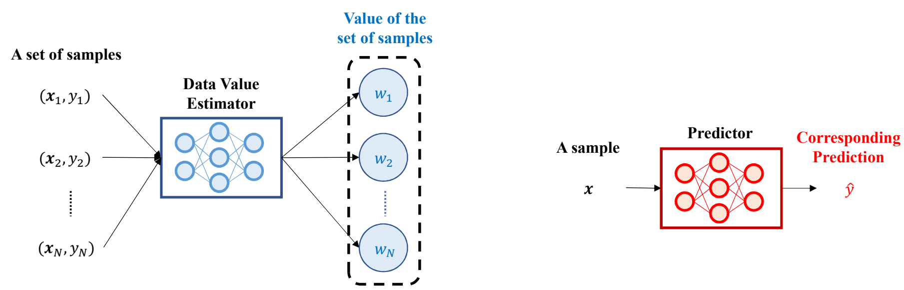
Supplementary Materials:
Data Valuation using Reinforcement Learning

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

1. Block diagrams for inference



(a) Data Valuation (b) Prediction

Figure 1. Block diagram of the proposed DVRL framework at inference time. (a) Data valuation, (b) Prediction. For data valuation, the input is a set of samples and the outputs are the corresponding data values. For prediction, the input is a sample and the output is the corresponding prediction. Both the data value estimator and predictor are fixed (not trained) at inference time.

2. Computational complexity

DVRL first trains the baseline model using the entire dataset (without re-weighting). Afterwards, we use this pre-trained baseline model to initialize the predictor network and apply fine-tuning with DVRL update steps. The convergence of the fine-tuning process is much faster than the convergence of training from the scratch.

We quantify the computational overhead of DVRL on the CIFAR-100 dataset (consisting 50k training samples and 100 label classes) with ResNet-32 (He et al., 2016) as a representative example. Overall, DVRL training takes less than 8 hours (given a pre-trained ResNet-32 model on the entire dataset) on a single NVIDIA Tesla V100 GPU without any hardware optimization. The pre-training time of ResNet-32 on the entire dataset (without re-weighting) is less than 4 hours; thus the total training time of DVRL is less than 12 hours from the scratch. On the other hand, the training time of Data Shapley (Ghorbani & Zou, 2019) (the most competitive benchmark) is more than a week on Fashion MNIST (consisting lower dimensional inputs and less number of classes) with a much simpler predictor model (2-layered CNNs).

At inference, the data value estimator can be used to obtain data value for each sample. The runtime of data valuation is typically much faster (less than 1 ms per sample) than the predictor model (e.g. ResNet-32).

3. Experimental details

In all experiments, we use Standard Normalizer to normalize the entire features to have zero mean and one standard deviation. We transform categorical variables into one-hot encoded embeddings. We use the inner iteration count N_I=200 for the predictor model, moving average window T=20, mini-batch size B_p=256 for the predictor model and mini-batch size

[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.

B_s=2000 for the DVE (large batch size often improves the stability of the reinforcement learning (McCandlish et al., 2018)). We use the learning rates β=0.01 for the DVE and α=0.001 for the predictor model. As the DVE architecture, for tabular datasets, we use 5-layer MLPs with 100 hidden units and ReLU activation function; and for image datasets, we use 5-layer MLPs with 100 hidden units and ReLU activation function on top of the CNN-based architecture used for the predictor model (such as ResNet-32 or WideResNet-28-10 (Zagoruyko & Komodakis, 2016)). The number of layers and hidden units are optimized with cross-validation.

4. Additional experimental results

4.1. Additional results on robust learning with noisy labels

We evaluate how DVRL can provide robustness for learning with noisy labels. We add various levels of label noise, ranging from 0% to 50%, to the training sets and evaluate how robust the proposed model (DVRL) is for the noisy dataset. In this experiment, we use three image datasets (CIFAR-10, Flower, and HAM 10000). Note that we initialize the predictor model using pre-trained Inception-v3 networks on ImageNet and only fine-tune the top layer (transfer learning setting).

Noise	CIFAR-10			Flower			HAM 10000		
ratio	*Clean*	DVRL	*Baseline*	*Clean*	DVRL	*Baseline*	*Clean*	DVRL	*Baseline*
0%	.8297	**.8305**	.8297	.9090	**.9292**	.9090	.7129	**.7148**	.7129
10%	.8281	**.8306**	.7713	.9057	**.9158**	.7441	.7094	**.7142**	.6746
20%	**.8285**	.8271	.6883	.9026	**.9152**	.5960	.7098	**.7126**	.6199
30%	**.8283**	.8262	.5897	.8889	**.8901**	.4546	**.7063**	.7005	.5508
40%	**.8259**	.8255	.4887	.8620	**.8787**	.2929	**.7028**	.6968	.4819
50%	**.8236**	.8225	.3832	.8542	**.8678**	.2962	**.7009**	.6814	.4132

Table 1. Robust learning results with various noise levels on CIFAR-10, Flower, and HAM 10000 datasets. *Clean* is the performance of the predictor model when it is only trained with the samples with clean labels (e.g. at 20% noise level, it uses only 80% clean samples). *Baseline* is the performance of the predictor model when it is trained with both noisy and clean labels.

Noisy labels significantly degrade the prediction performance when they are included in the training dataset (see the increasing performance differences between *Baseline* and *Clean* in Table 1). DVRL demonstrates high robustness up to high noisy label ratio (50%). In some cases (even without noisy labels (i.e. 0% noise ratio)), the prediction performance even outperforms the *Clean* case, as DVRL prioritizes some clean samples more than others. Overall, DVRL framework is promising in maintaining high prediction performance even with a significant increase in the amount of noisy labels.

4.2. Additional results on robust learning with noisy features

σ	Blog		Adult	
	Baseline	DVRL	*Baseline*	DVRL
0.1	0.733	**0.819**	0.802	**0.820**
0.2	0.647	**0.798**	0.753	**0.788**
0.3	0.626	**0.766**	0.699	**0.771**
0.4	0.623	**0.717**	0.652	**0.725**

Table 2. Testing accuracy when trained with noisy features. σ is the standard deviation of the added Gaussian noise, quantifying the level of perturbation on the features.

In this section, we consider training with noisy input features, with a clean validation set. We add Gaussian noise with zero mean and a certain standard deviation of σ to each feature in the training set independently. We use two tabular datasets (Adult and Blog) to evaluate the robustness of DVRL on input noise. As can be seen in Table 2, DVRL is robust with noise on the features and the performance gains are higher with larger noise in comparison to *Baseline* (i.e. treat all the noisy training samples equally), since DVRL can discover the training samples with less corrupted by the additive noise among the entire noisy training samples and provide higher weights on those less noisy samples.

4.3. Additional results on corrupted sample discovery with 20% label noise

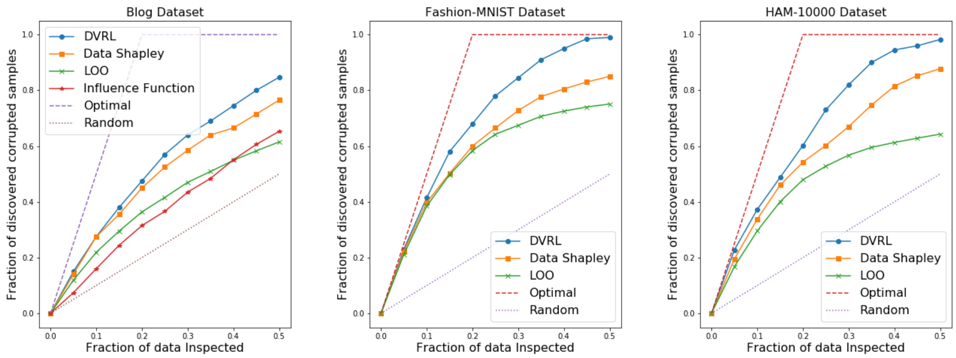


Figure 2. Discovering corrupted samples in three datasets ((a) Blog, (b) Fashion-MNIST, (c) HAM 10000 datasets) in the presence of 20% noisy labels. 'Optimal' saturates at the 20 % of the fraction, perfectly assigning the lowest data value scores to the samples with noisy labels. 'Random' does not introduce any knowledge on distinguishing clean vs. noisy labels, and thus the fraction of discovered corrupt samples is proportional to the amount of inspection.

4.4. Additional results on removing high/low value samples with 20% label noise

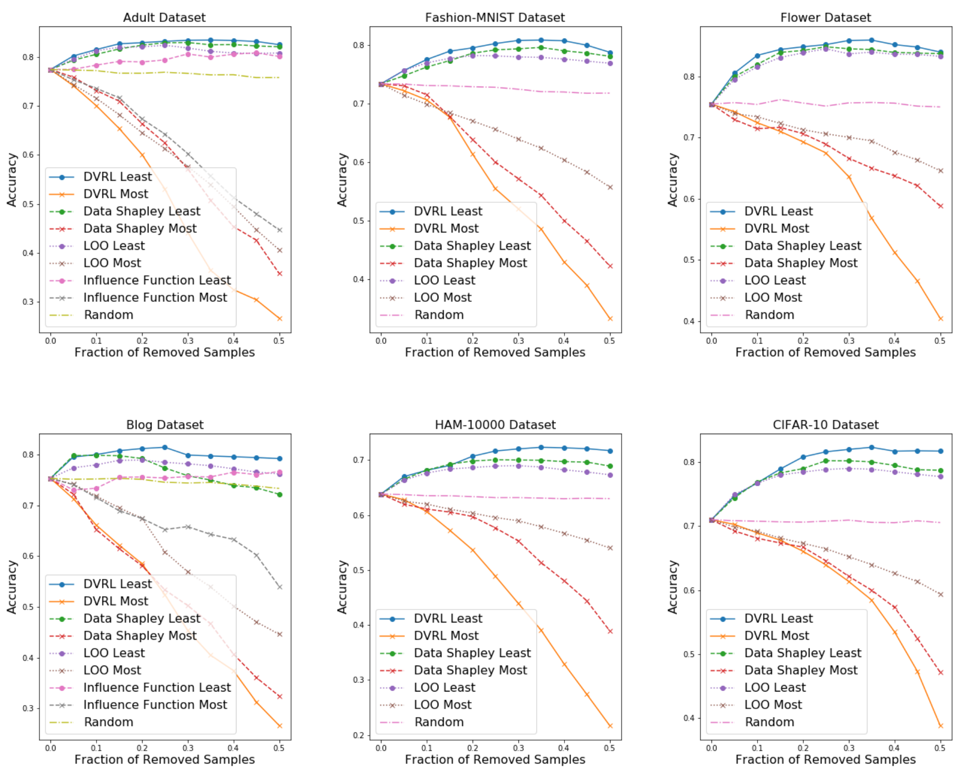


Figure 3. Performance after removing the most (marked with $\times$) and least (marked with $\bigcirc$) important samples according to the estimated data values. We assume a label noise with 20% ratio on (a) Adult, (b) Fashion-MNIST, (c) Flower, (d) Blog, (e) HAM 10000, (f) CIFAR-10 datasets.

In this subsection, we focus on removing high/low value samples in the presence of label noise in the training data. As noisy samples hurt the prediction performance, an optimal DVE with a clean validation dataset should assign low values to the noisy samples. With the removal of samples with noisy labels ('Least' setting), the prediction performance should either increase, or at least decrease much slower, compared to removal of samples with correct labels ('Most' setting). In this experiment, we introduce label noise to 20% of the samples by replacing true labels with random labels. As shown in Fig. 3, for all data valuation methods, the prediction performance tends to first slowly increase and then decrease in the 'Least' setting; and tends to rapidly decrease in the 'Most' setting. Yet, DVRL achieves the slowest performance decrease in 'Least' setting and the fastest performance decrease in the 'Most' setting, reflecting its superiority in data valuation.

5. Learning curves of DVRL

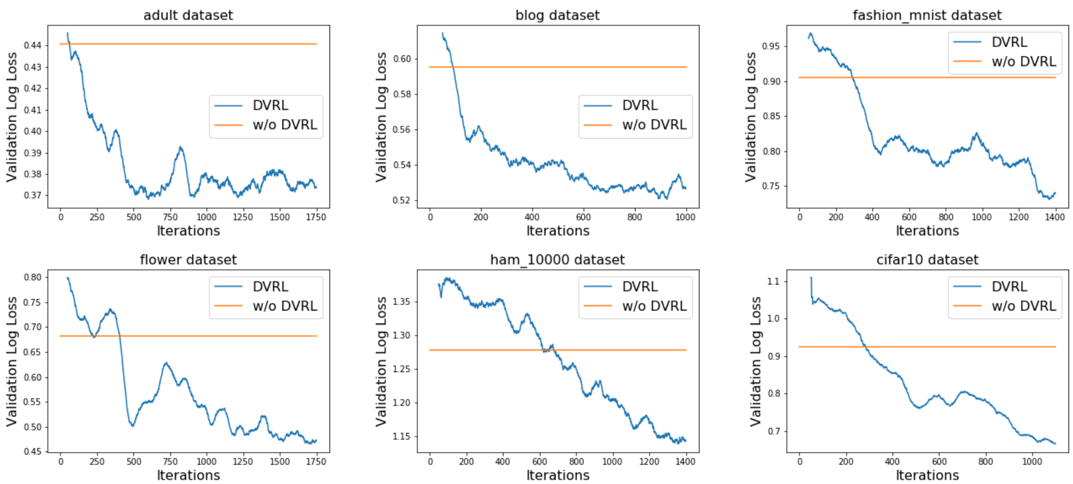


Figure 4. Learning curves of DVRL for 6 datasets with 20% noisy labels. x-axis: the number of iterations for data value estimator training, y-axis: validation performance (log loss). (Orange: validation log loss without DVRL, Blue: validation log loss with DVRL)

Fig. 4 shows the learning curves of DVRL on the noisy data (with 20% label noise) setting in comparison to the validation log loss without DVRL (directly trained on the noisy data without re-weighting) on 2 tabular datasets (Adult and Blog) and 4 image datasets (Fashion-MNIST, Flower, HAM 10000, and CIFAR-10).

6. Confidence intervals of DVRL performance on corrupted sample discovery experiments

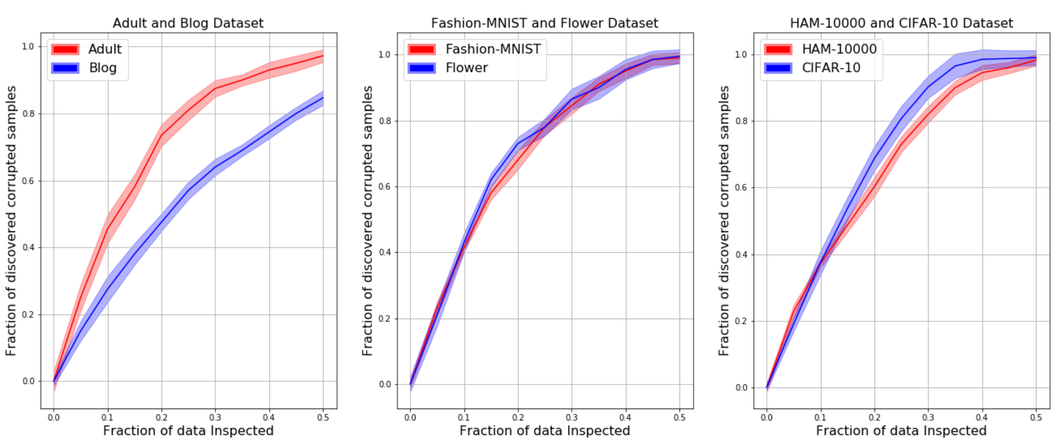


Figure 5. Corrupted sample discovery performance with 95% confidence intervals (computed by 10 independent runs) according to the estimated data values by DVRL. We assume a label noise with 20% ratio on (a) Adult and Blog, (b) Fashion-MNIST and Flower (c) HAM 10000 and CIFAR-10 datasets.

Fig. 5 shows the confidence intervals of DVRL, demonstrating its high stability against randomness of the initialization or data shuffling.