1. Appendix

1.1. Proof for Theorem 1

Theorem 1. *If the third-derivative of the loss function at θ^* is sufficiently small, the second-order group influence function (denoted by $\mathcal{I}^{(2)}(\mathcal{U})$) when all samples in a group $\mathcal{U}$ are up-weighted by ϵ is:*

$$\mathcal{I}^{(2)}(\mathcal{U}) = \mathcal{I}^{(1)}(\mathcal{U}) + \mathcal{I}^{'}(\mathcal{U}) \tag{1}$$

where:

$$\mathcal{I}^{(1)}(\mathcal{U}) = -\frac{1}{1-p}\frac{1}{|\mathcal{S}|}H_{\theta^*}^{-1}\sum_{z\in\mathcal{U}}\nabla\ell(h_{\theta^*}(z))$$

and

$$\mathcal{I}^{'}(\mathcal{U}) = \frac{p}{1-p}\Big(I - (\nabla^2 L_\emptyset(\theta^*))^{-1}\frac{1}{|\mathcal{U}|}\sum_{z\in\mathcal{U}}\nabla^2\ell(h_{\theta^*}(z))\Big)\theta^{(1)}$$

Proof. We consider the empirical risk minimization problem where the learning algorithm is given training samples $\mathcal{S} = \{z_i = (x_i, y_i)\}_{i=1}^m$ drawn i.i.d from some distribution $\mathcal{P}$. Let Θ be the space of the parameters and h_θ be the hypothesis to learn. The goal is to select model parameters θ to minimize the empirical risk as follows:

$$\min_{\theta\in\Theta}\ L_\emptyset(\theta) := \frac{1}{|\mathcal{S}|}\sum_{z\in\mathcal{S}}\ell(h_\theta(z)), \tag{2}$$

The ERM problem with a subset $\mathcal{U}\subset\mathcal{S}$ removed from $\mathcal{S}$ is as follows:

$$L_\mathcal{U}(\theta) := \frac{1}{|\mathcal{S}|-|\mathcal{U}|}\sum_{z\in\mathcal{S}\setminus\mathcal{U}}\ell(h_\theta(z)) \tag{3}$$

In our formulation for the group influence function, we assume that weights of samples in the set $\mathcal{U}$ has been up-weighted all by ϵ. This leads to a down-weighting of the remaining training samples by $\tilde{\epsilon} = \frac{|\mathcal{U}|}{|\mathcal{S}|-|\mathcal{U}|}\epsilon$, to conserve the empirical weight distribution of the training data. We denote p as $\frac{|\mathcal{U}|}{|\mathcal{S}|}$ to denote the fraction of up-weighted training samples. Therefore, the resulting ERM optimization can be written as:

$$\theta_\mathcal{U}^\epsilon = \arg\min_\theta L_\mathcal{U}^\epsilon(\theta) \tag{4}$$

where:

$$L_\mathcal{U}^\epsilon(\theta) = L_\emptyset(\theta) + \frac{1}{|\mathcal{S}|}\Big(\sum_{z\in\mathcal{S}\setminus\mathcal{U}} -\tilde{\epsilon}\ell(h_\theta(z)) + \sum_{z\in\mathcal{U}}\epsilon\ell(h_\theta(z))\Big) \tag{5}$$

and $\tilde{\epsilon} = \frac{|\mathcal{U}|\epsilon}{|\mathcal{S}|-|\mathcal{U}|}$. We consider the stationary condition where the gradient of $L_\mathcal{U}^\epsilon$ is zero. More specifically:

$$0 = \nabla L_\mathcal{U}^\epsilon(\theta_\mathcal{U}^\epsilon) = \nabla L_\emptyset(\theta_\mathcal{U}^\epsilon) + \frac{1}{|\mathcal{S}|}\Big(-\tilde{\epsilon}\sum_{z\in\mathcal{S}\setminus\mathcal{U}}\nabla\ell(h_{\theta_\mathcal{U}^\epsilon}(z)) + \epsilon\sum_{z\in\mathcal{U}}\nabla\ell(h_{\theta_\mathcal{U}^\epsilon}(z))\Big) \tag{6}$$

Next we expand $\nabla L_\emptyset(\theta_\mathcal{U}^\epsilon)$ around the optimal parameter θ^* using Taylor's expansion and retrieve the terms with coefficients $\mathcal{O}(\epsilon)$ to find $\theta^{(1)}$:

$$\begin{aligned} 0 = {} & \nabla L_\emptyset(\theta^*) + \nabla^2 L_\emptyset(\theta^*)(\theta_\mathcal{U}^\epsilon - \theta^*) \\ & + \frac{1}{|\mathcal{S}|}\Big(-\tilde{\epsilon}\sum_{z\in\mathcal{S}}\nabla\ell(h_{\theta^*}(z))\Big) \\ & + \frac{1}{|\mathcal{S}|}(\epsilon+\tilde{\epsilon})\sum_{z\in\mathcal{U}}\nabla\ell(h_{\theta^*}(z)) \end{aligned} \tag{7}$$

At the optimal parameters θ^*, $\nabla L_\emptyset(\theta^*) = 0$ and $\theta_\mathcal{U}^\epsilon - \theta^* = \epsilon\theta^{(1)}$, thus simplifying Equation (7):

$$\begin{aligned} & \epsilon\nabla^2 L_\emptyset(\theta^*)\theta^{(1)} \\ = {} & \frac{1}{|\mathcal{S}|}(\tilde{\epsilon}\sum_{z\in\mathcal{S}\setminus\mathcal{U}} - \epsilon\sum_{z\in\mathcal{U}})\nabla\ell(h_{\theta^*}(z)) \\ = {} & \tilde{\epsilon}\nabla L_\emptyset(\theta^*) - \frac{1}{|\mathcal{S}|}(\tilde{\epsilon}+\epsilon)\sum_{z\in\mathcal{U}}\nabla\ell(h_{\theta^*}(z)) \\ = {} & -\frac{1}{|\mathcal{S}|}(\tilde{\epsilon}+\epsilon)\sum_{z\in\mathcal{U}}\nabla\ell(h_{\theta^*}(z)) \\ = {} & -\frac{1}{|\mathcal{S}|}\frac{|\mathcal{S}|}{(|\mathcal{S}|-|\mathcal{U}|)}\epsilon\sum_{z\in\mathcal{U}}\nabla\ell(h_{\theta^*}(z)) \end{aligned} \tag{8}$$

Substituting $|\mathcal{U}|/|\mathcal{S}|$ as p, we get the following identity:

$$\theta^{(1)} = -\frac{1}{|\mathcal{S}|}\frac{1}{1-p}\Big(\nabla^2 L_\emptyset(\theta^*)\Big)^{-1}\sum_{z\in\mathcal{U}}\nabla\ell(h_{\theta^*}(z)) \tag{9}$$

where $\theta^{(1)}$ is the first-order approximation of group influence function. We denote the first-order approximation as $\mathcal{I}^{(1)}$.

Next we derive $\mathcal{I}^{'}$ by comparing terms to the order of $\epsilon^{(2)}$ (i.e. $\mathcal{O}(\epsilon^2)$) in Equation (6) by expanding around θ^*:

$$\begin{aligned} 0 = {} & \nabla^2 L_\emptyset(\theta^*)(\epsilon\theta^{(1)} + \epsilon^2\theta^{(2)} + \cdots) \\ & + \frac{1}{|\mathcal{S}|}(-\tilde{\epsilon}\sum_{z\in\mathcal{S}}\nabla\ell(h_{\theta_\mathcal{U}^\epsilon}(z)) \\ & + \frac{|\mathcal{S}|\epsilon}{|\mathcal{S}|-|\mathcal{U}|}\sum_{z\in\mathcal{U}}\nabla\ell(h_{\theta_\mathcal{U}^\epsilon}(z))) \end{aligned} \tag{10}$$

where:

$$\nabla \ell(h_{\theta^\epsilon_{\mathcal{U}}}(z)) = \nabla \ell(h_{\theta^*}(z)) + \nabla^2 \ell(h_{\theta^*}(z))(\theta^\epsilon_{\mathcal{U}} - \theta^*) + \cdots \tag{11}$$

Substituting Equation (11) in Equation (10), and expanding in $\mathcal{O}(\epsilon^2)$ we get the following identity:

$$\nabla^2 L_\emptyset(\theta^*)\epsilon^2\theta^{(2)} - \frac{1}{|\mathcal{S}|}\frac{|\mathcal{U}|}{|\mathcal{S}| - |\mathcal{U}|}\sum_{z\in\mathcal{S}} \nabla^2 \ell(h_{\theta^*}(z))\epsilon^2\theta^{(1)} \tag{12}$$

$$+\frac{1}{|\mathcal{S}| - |\mathcal{U}|}\sum_{z\in\mathcal{U}} \nabla^2 \ell(h_{\theta^*}(z))\epsilon^2\theta^{(1)}$$

Taking the common coefficients $|\mathcal{U}|/(|\mathcal{S}| - |\mathcal{U}|)$ out and rearranging Equation (12), we get the following identity:

$$\nabla^2 L_\emptyset(\theta^*)\theta^{(2)} = \frac{|\mathcal{U}|}{|\mathcal{S}| - |\mathcal{U}|}\Big(\frac{1}{|\mathcal{S}|}\sum_{z\in\mathcal{S}} \nabla^2 \ell(h_{\theta^*}(z)) \tag{13}$$

$$-\frac{1}{|\mathcal{U}|}\sum_{z\in\mathcal{U}} \nabla^2 \ell(h_{\theta^*}(z))\Big)\theta^{(1)}$$

Now we multiply both sides of the Equation (13) with the Hessian inverse i.e. $\nabla^2 L_\emptyset(\theta^*)^{-1}$, we obtain the cross-term involving the gradients and the Hessians of the removed points in the second order influence function as follows:

$$\theta^{(2)} = \frac{p}{1-p}\Big(I - \frac{1}{|\mathcal{U}|}\nabla^2\Big(L_\emptyset(\theta^*)\Big)^{-1}\sum_{z\in\mathcal{U}} \nabla^2 \ell(h_{\theta^*}(z))\Big)\theta^{(1)} \tag{14}$$

where $p = |\mathcal{U}|/|\mathcal{S}|$ and we denote $\theta^{(2)}$ as $\mathcal{I}^{'}$. We combine Equation (9) and (14), to write the second order influence function $\mathcal{I}^{(2)}(\mathcal{U})$ as:

$$\mathcal{I}^{(2)}(\mathcal{U}) = \mathcal{I}^{(1)}(\mathcal{U}) + \mathcal{I}^{'}(\mathcal{U}) \tag{15}$$

□

1.2. Synthetic Dataset - Influence Computation

The synthetic data is sampled from a multivariate Gaussian distribution with 5 dimensions and consists of two classes. We sample a total of 10000 points for our experiments. For the first class, the mean is kept at 0.1, while for the second class the mean is kept at 0.8. The covariance matrix in both the classes is a diagonal matrix whose entries are sampled randomly between 0 and 1.

1.3. Coherent Groups

In case of randomly removed groups, the perturbations to the model are relatively small when compared to a case where groups having similar properties are removed. This is also the experimental setting in (Koh, Ang, Teo, and Liang, 2019), where the first-order group influence function is analysed. In our experimental setup for the MNIST dataset, we remove groups of sizes (denoted by $\|\mathcal{U}\|$) ranging from 100 to 3500 from a specific class. Essentially when a group of points are removed, they are ensured to be from a similar class.

1.4. Synthetic Dataset - Group Selection

For testing our optimal group selection procedure for second-order group influence functions we generate synthetic data consisting of 20000 samples from `sklearn.datasets.make_blobs` consisting of 5 features from 4 distinct classes. The generated data is in the form of isotropic Gaussian blobs.

1.5. Plots for Neural Network Experiments

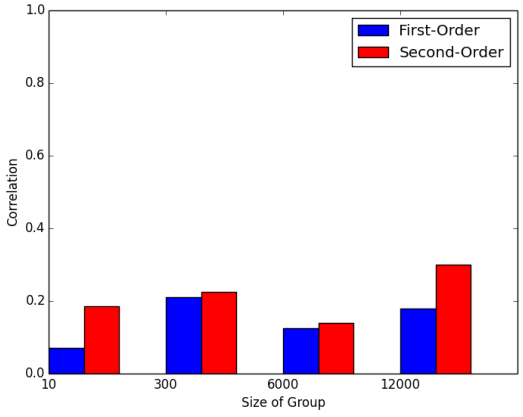

Figure 1. Correlation vs group size plot for neural networks (MNIST dataset).

In case of neural networks, we use the existing first-order approximation of influence functions and our proposed second-order influence functions to approximate effect of leave-k-out retraining on the test-loss for a particular sample. We use the MNIST dataset in our experiments. Across different groups sizes ranging from 10% to 50% of the entire training data, we observe that the correlation with the ground-truth is far from significant (Figure (1)). Up-weighting of a group leads to a large perturbation to the underlying model, where a local approximation via influence functions around the loss function at the optimal parameters might not lead to an accurate estimate in case of deep networks when compared to a linear model. A comprehensive investigation of the behaviour of influence functions in deep learning is a direction for future work.

1.6. On Infinitesimal Jackknife and Influence Functions

Infinitesimal jackknife and influence functions are methods from robust statistics to linearly approximate cross-validation procedures and attaching standard errors to point estimates (Efron, 1992; Jaeckel, 1972; Cook and Weisberg, 1980; Cook and Sanford, 1982). Previously (Efron, 1992) have shown that linear approximations for cross-validation in case of infinitesimal jackknife and influence functions are relatively similar, although the technical ideas are different. Recently (Giordano, Stephenson, Liu, Jordan, and Broderick, 2018) approximated cross-validation procedures by using a linear approximation to the dependence of the cross-validation fitting procedure on the empirical weight distribution of the training data in the ERM. This procedure is similar in idea to (Koh and Liang, 2017), but differs technically where a training sample was up-weighted by an ϵ factor and the approximated solution to the modified ERM was found using influence functions. However in both the cases of (Giordano et al., 2018) and (Koh and Liang, 2017; Koh et al., 2019) higher-order terms have been ignored. A recent work (Giordano, Jordan, and Broderick, 2019) focuses on higher-order terms of infinitesimal jackknife for leave-k-out retraining approximations. (Giordano et al., 2019) specifically provide tools for extending infinitesimal jackknife(Giordano et al., 2018) to higher-order terms with finite sample accuracy bounds. We give a small description on how the higher-order terms in (Giordano et al., 2019) have been derived, in the following part:

With a training set $\{z_i = (x_i, y_i)\}_{i=1}^N$ we consider the following ERM problem:

$$L(\theta, w) = \frac{1}{N}\sum_{i=1}^{N} w_i \ell(z_i, \theta) \tag{16}$$

and let $\hat{\theta}(w)$ be the solution to the following optimization problem:

$$\hat{\theta}(w) = \arg\min_{\theta} L(\theta, w) \tag{17}$$

Let $\hat{\theta}$ be the solution to the ERM problem with $w_i = 1, \quad \forall i \in [1, N]$. We denote this vector of w in case of $\hat{\theta}$ as 1_N. We consider the change in the empirical weight distribution of the training data in the ERM as $\Delta w = w - 1_N$. Infinitesimal jackknife procedures compute the directional derivative of $\hat{\theta}(w)$ in a direction denoted by Δw. Let $\delta_w^1\hat{\theta}(w)$ be the first-order directional derivative of $\hat{\theta}(w)$ in the direction Δw, $\delta_w^2\hat{\theta}(w)$ be the second-order derivative and $\delta_w^k\hat{\theta}(w)$ be the k^{th} order derivative. As a representative example, $\delta_w^1\hat{\theta}(w)$ is defined in the following way:

$$\delta_w^1\hat{\theta}(w) = \sum_{n=1}^{N} \frac{\partial\hat{\theta}(w)}{\partial w_n}|_w \Delta w_n \tag{18}$$

and the second-order directional derivative as:

$$\delta_w^2\hat{\theta}(w) = \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} \frac{\partial^2\hat{\theta}(w)}{\partial w_{n_1}\partial w_{n_2}}|_w \Delta w_{n_1}\Delta w_{n_2} \tag{19}$$

Infinitesimal jackknife focuses on finding the solution to $\hat{\theta}(w)$ by a Taylor series approximation around $\hat{\theta}$ as follows :

$$\hat{\theta}(w) = \hat{\theta}(1_N) + \sum_{i=1}^{k} \frac{1}{i!}\delta_w^i\hat{\theta}(1_N) \tag{20}$$

(Giordano et al., 2019) specifically provide tools to obtain the higher-order directional derivatives i.e ($k \geq 2$) through a recursive procedure and also provide their associated finite sample accuracy bounds. However any experiment supporting how their derived higher-order infinitesimal jackknife behave in case of practical machine learning models was not explored in (Giordano et al., 2019). Note that even in influence functions $\hat{\theta}$ is used as the known solution around which the up-weighted solution of the ERM is found. However the technical derivation of the parameters of the ERM with up-weighted training examples is different as can be observed in (Cook and Weisberg, 1980; Koh and Liang, 2017; Koh et al., 2019). Our work specifically focuses on higher-order terms in case of influence functions first shown in (Cook and Weisberg, 1980) and subsequently by (Koh and Liang, 2017; Koh et al., 2019). We use a combination of perturbation series and Taylor series to compute the higher-order terms in case of influence functions. In the process we make certain practical modifications like conserving the empirical weight distribution of the training data to one, which is particularly important as shown in the main paper. Note that higher-order terms of influence functions as shown in this paper is different from the higher-order terms of infinitesimal jackknife as shown in (Giordano et al., 2019) due to these differences. Also we show practically how the second-order influence function is beneficial over existing first-order influence function in case of approximating leave-k-out retraining procedures for machine learning models especially for large values of k as well as for influential group selection procedure.

2. Efficient Computation of Second-Order Influence Function

The second-order group influence function ($\mathcal{I}^{(2)}(\mathcal{U}, z_t)$) can be expressed as:

$$\nabla\ell(h_{\theta^*}(z_t))^T \Big\{ \underbrace{\frac{1}{|\mathcal{S}|}\frac{1-2p}{(1-p)^2} H_{\theta^*}^{-1} \sum_{z\in\mathcal{U}} \nabla\ell(h_{\theta^*}(z))}_{Term1} +$$

$$\underbrace{\frac{1}{(1-p)^2}\frac{1}{|\mathcal{S}|^2}\sum_{z\in\mathcal{U}} H_{\theta^*}^{-1}\nabla^2\ell(h_{\theta^*}(z))H_{\theta^*}^{-1}\sum_{z'\in\mathcal{U}}\nabla\ell(h_{\theta^*}(z'))\Big\}}_{Term2} \tag{21}$$

$Term1$ can be efficiently computed using a combination of Hessian-vector product (Pearlmutter, 1994) and conjugate-gradient. $Term2$, which captures cross-dependencies amongst the samples can also be computed efficiently using three steps. First $v_1 = H_{\theta^*}^{-1}\sum_{z'\in\mathcal{U}}\nabla\ell(h_{\theta^*}(z'))$ is computed using a combination of Hessian-vector product and conjugate gradient. In the next step, $v_2 = \nabla^2\ell(h_{\theta^*}(z))v_1$ is computed using Hessian-vector product and the output of this step is a vector. In the final step, $v_3 = H_{\theta^*}^{-1}v_2$ is computed using Hessian-vector product and conjugate-gradient. The second-order group influence function thus requires an extra computation of conjugate-gradient. While the second-order group influence function is a bit more expensive than the first-order one (due to an extra CG operation), it is much faster than retraining to compute the ground-truth influence. For example, the running time of first-order influence on MNIST for different groups (up to 20 % of removed data) averaged over 100 runs is 16.7 $\pm$ 2.56s while that of the second-order method is 28.2 $\pm$ 4.8s.

3. Additional Experimental Results

In this section, we provide an additional experimental result with the second-order group influence function evaluated on the Iris dataset for logistic regression. From Figure (2), we observe that the second-order group influence estimates have a better correlation with the ground-truth across different group sizes. Specifically we notice that when more than 33% of the training data is removed, the improvement in correlation by the second-order influence function is more. In this experimental setup, the groups were removed randomly. Specifically for each group size, 200 groups were randomly removed and the experiment was repeated for 10 trials to obtain the mean correlation.

4. Experimental Details on MNIST

For the experiments with the MNIST dataset, for each group size and test-point (in case of both the random and coherent groups), 500 groups were removed and the experiment is repeated for 10 trials. In our paper, we report the mean correlation amongst 10 experimental trials.

5. Relationship To Representer Point Theorem

Identification of influential samples in the training set can also be done by alternative methods such as representer point theorem (Yeh, Kim, Yen, and Ravikumar, 2018). In particular, unlike influence functions, (Yeh et al., 2018) assigns an influence score by kernel evaluations at the training samples. While both (Yeh et al., 2018) and our method explain a test-prediction through the lens of the training data, the definitions of influences or importances are different in both cases. Influence functions define importance analogous to the leave-out re-training procedure while the kernel function defined in (Yeh et al., 2018) evaluates influences by relying on the weighted sum of the feature similarity of the training samples in the pre-activation layer of a deep network. Investigation of the exact relationship between kernel based methods like representer theorem and influence functions is a direction for future work.

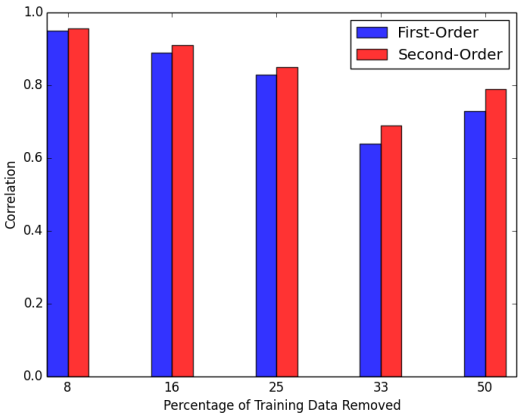

Figure 2. Correlation vs group size plot for Iris dataset.

References

R. Cook and S. Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980. ISSN 0040-1706. doi: 10.1080/00401706.1980.10486199.

R. D. Cook and W. Sanford. Residuals and influence in regression. *Chapman and Hall*, 1982. URL http://hdl.handle.net/11299/37076.

B. Efron. Jackknife-after-bootstrap standard errors and influence functions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(1):83–127, 1992. ISSN 00359246. URL http://www.jstor.org/stable/2345949.

R. Giordano, W. Stephenson, R. Liu, M. I. Jordan, and T. Broderick. A swiss army infinitesimal jackknife. In *AISTATS*, 2018.

R. Giordano, M. I. Jordan, and T. Broderick. A higher-order swiss army infinitesimal jackknife. *ArXiv*, abs/1907.12116, 2019.

L. A. Jaeckel. The infinitesimal jackknife. *Technical Report*, 1:1–35, 06 1972.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/koh17a.html.

P. W. Koh, K. Ang, H. H. K. Teo, and P. Liang. On the accuracy of influence functions for measuring group effects. *CoRR*, abs/1905.13289, 2019. URL http://arxiv.org/abs/1905.13289.

B. A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Comput.*, 6(1):147–160, Jan. 1994. ISSN 0899-7667. doi: 10.1162/neco.1994.6.1.147. URL http://dx.doi.org/10.1162/neco.1994.6.1.147.

C. Yeh, J. S. Kim, I. E. Yen, and P. Ravikumar. Representer point selection for explaining deep neural networks. *CoRR*, abs/1811.09720, 2018. URL http://arxiv.org/abs/1811.09720.