Discriminative Jackknife: Quantifying Uncertainty in Deep Learning via Higher-Order Influence Functions

Ahmed M. Alaa [1] **Mihaela van der Schaar** [1 2]

Appendix

A. Influence Functions: Background & Key Concepts

A.1. Formal Definition

Robust statistics is the branch of statistics concerned with the detection of outlying observations. An estimator is deemed *robust* if it produces similar results as the majority of observations indicates, regardless of how a minority of other observations is perturbed ((Huber & Ronchetti, 1981)). The influence function measures these effects in statistical functionals by analyzing the behavior of a functional not only at the distribution of interest, but also in an entire neighborhood of distributions around it. Lack of model robustness is a clear indicator of model uncertainty, and hence influence functions arise naturally in our method as a (pointwise) surrogate measure of model uncertainty. In this section we formally define influence functions and discuss its properties.

The pioneering works in ((Hampel et al., 2011)) and ((Huber & Ronchetti, 1981)) coined the notion of influence functions to assess the robustness of statistical functionals to perturbations in the underlying distributions. Consider a statistical functional $T : \mathcal{P} \rightarrow \mathbb{R}$, defined on a probability space $\mathcal{P}$, and a probability distribution $\mathbb{P} \in \mathcal{P}$. Consider distributions of the form $\mathbb{P}_{\varepsilon,z} = (1-\varepsilon)\mathbb{P} + \varepsilon\Delta z$ where Δz denotes the Dirac distribution in the point $z = (\boldsymbol{x}, y)$, representing the contaminated part of the data. For the functional T to be considered robust, $T(\mathbb{P}_{\varepsilon,z})$ should not be too far away from $T(\mathbb{P})$ for any possible z and any small ε. The limiting case of $\varepsilon \rightarrow 0$ defines the influence function. That is, Then the influence function of T at $\mathbb{P}$ in the point z is defined as

$$\mathcal{I}(z;\mathbb{P}) = \lim_{\varepsilon \rightarrow 0} \frac{T(\mathbb{P}_{\varepsilon,z}) - T(\mathbb{P})}{\varepsilon} \triangleq \frac{\partial}{\partial \varepsilon} T(\mathbb{P}_{\varepsilon,z}) \Big|_{\varepsilon=0}, \tag{1}$$

The influence function measures the robustness of T by quantifying the effect on the estimator T when adding an infinitesimally small amount of contamination at the point z. If the supremum of $\mathcal{I}(.)$ over z is bounded, then an infinitesimally small amount of perturbation cannot cause arbitrary large changes in the estimate. Then small amounts of perturbation cannot completely change the estimate which ensures the robustness of the estimator.

A.2. The von Mises Expansion

The von Mises expansion is a distributional analog of the Taylor expansion applied for a functional instead of a function. For two distributions $\mathbb{P}$ and $\mathbb{Q}$, the Von Mises expansion is ((Fernholz, 2012)):

$$T(\mathbb{Q}) = T(\mathbb{P}) + \int \mathcal{I}^{(1)}(z;\mathbb{P})\, d(\mathbb{Q}-\mathbb{P}) + \frac{1}{2}\int \mathcal{I}^{(2)}(z;\mathbb{P})\, d(\mathbb{Q}-\mathbb{P}) + \ldots, \tag{2}$$

where $\mathcal{I}^{(k)}(z;\mathbb{P})$ is the k^{th} order influence function. By setting $\mathbb{Q}$ to be a perturbed version of $\mathbb{P}$, i.e., $\mathbb{Q} = \mathbb{P}_{\varepsilon}$, the von Mises expansion at point z reduces to:

$$T(\mathbb{P}_{\varepsilon,z}) = T(\mathbb{P}) + \varepsilon\mathcal{I}^{(1)}(z;\mathbb{P}) + \frac{\varepsilon^2}{2}\mathcal{I}^{(2)}(z;\mathbb{P}) + \ldots, \tag{3}$$

[1]UCLA [2]Cambridge University. Correspondence to: Ahmed M. Alaa <ahmedmalaa@ucla.edu>.

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

and so the k^{th} order influence function is operationalized through the derivative

$$\mathcal{I}^{(k)}(z;\mathbb{P}) \triangleq \frac{\partial}{\partial^k \varepsilon} T(\mathbb{P}_{\varepsilon,z})\Big|_{\varepsilon=0}. \tag{4}$$

A.3. Influence Function of Model Loss

Now we apply the mathematical definitions in Sections A.1 and A.2 to our learning setup. In our setting, the functional $T(.)$ corresponds to the (trained) model parameters $\hat{\theta}$ and the distribution $\mathbb{P}$. In this case, influence functions of $\hat{\theta}$ computes how much the model parameters would change if the underlying data distribution was perturbed infinitesimally.

$$\mathcal{I}_\theta^{(1)}(z) = \frac{\partial\, \hat{\theta}_{\varepsilon,z}}{\partial\, \varepsilon}\Big|_{\varepsilon=0}, \quad \hat{\theta}_{\varepsilon,z} \triangleq \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \ell(z_i;\theta) + \varepsilon\, \ell(z;\theta). \tag{5}$$

Recall that in the definition of the influence function $\mathbb{P}_{\varepsilon,z} = (1-\varepsilon)\mathbb{P} + \varepsilon\Delta z$ where Δz denotes the Dirac distribution in the point $z = (\boldsymbol{x}, y)$. Thus, the (first-order) influence function in (5) corresponds to perturbing a training data point z by an infinitesimally small change ε and evaluating the corresponding change in the learned model parameters $\hat{\theta}$. More generally, the k^{th} order influence function of $\hat{\theta}$ is defined as follows:

$$\mathcal{I}_\theta^{(k)}(z) = \frac{\partial^k\, \hat{\theta}_{\varepsilon,z}}{\partial\, \varepsilon^k}\Big|_{\varepsilon=0}. \tag{6}$$

By applying the von Mises expansion, we can approximate the parameter of a model trained on the training dataset with perturbed data point z as follows:

$$\hat{\theta}_{\varepsilon,z} \approx \hat{\theta} + \varepsilon\, \mathcal{I}_\theta^{(1)}(z) + \frac{\varepsilon^2}{2}\mathcal{I}_\theta^{(2)}(z) + \ldots + \frac{\varepsilon^m}{m!}\mathcal{I}_\theta^{(m)}(z), \tag{7}$$

where m is the number of terms included in the truncated expansion. When $m = \infty$, the exact parameter $\hat{\theta}_{\varepsilon,z}$ without the need to re-train the model.

A.4. Connection to leave-one-out estimators

Our uncertainty estimator depends on perturbing the model parameters by removing a single training point at a time. Note that removing a point z is the same as perturbing z by $\varepsilon = \frac{-1}{n}$, hence we obtain an ($m^{th}$ order) approximation of the parameter change due to removing the point z as follows:

$$\hat{\theta}_{-z} - \hat{\theta} \approx \frac{-1}{n}\mathcal{I}_\theta^{(1)}(z) + \frac{1}{2n^2}\mathcal{I}_\theta^{(2)}(z) + \ldots + \frac{(-1)^m}{n^m \cdot m!}\mathcal{I}_\theta^{(m)}(z), \tag{8}$$

where $\hat{\theta}_{-z}$ is the model parameter learned by removing the data point z from the training data.

B. Derivation of Influence Functions

Recall that the LOO parameter $\hat{\theta}_{i,\epsilon}$ is obtained by solving the optimization problem:

$$\hat{\theta}_{i,\epsilon} = \arg\min_{\theta\in\Theta} L(\mathcal{D},\theta) + \epsilon\cdot\ell(y_i, f(\boldsymbol{x}_i;\theta)). \tag{9}$$

Let us first derive the first order influence function $\mathcal{I}^{(1)}(\boldsymbol{x}_i, y_i)$. Let us first define $\Delta_{i,\epsilon} \triangleq \hat{\theta}_{i,\epsilon} - \hat{\theta}$. The first order influence function is given by:

$$\mathcal{I}^{(1)}(\boldsymbol{x}_i, y_i) = \frac{\partial\hat{\theta}_{i,\epsilon}}{\partial\epsilon} = \frac{\partial\Delta_{i,\epsilon}}{\partial\epsilon}. \tag{10}$$

Note that, since $\hat{\theta}_{i,\epsilon}$ is the minimizer of (9), then the perturbed loss has to satisfy the following (first order) optimality condition:

$$\nabla_\theta \left\{L(\mathcal{D},\theta) + \epsilon\cdot\ell(y_i, f(\boldsymbol{x}_i;\theta))\right\}\big|_{\theta=\hat{\theta}_{i,\epsilon}} = 0. \tag{11}$$

Since $\lim_{\epsilon \to 0} \hat{\theta}_{i,\epsilon} = \hat{\theta}$, then we can write the following Taylor expansion:

$$\nabla_\theta \sum_{k=0}^{\infty} \frac{\Delta_{i,\epsilon}^k}{k!} \cdot \nabla_\theta^k \left\{ L(\mathcal{D}, \hat{\theta}) + \epsilon \cdot \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\} = 0. \tag{12}$$

Now by dropping the $o(\|\Delta_{i,\epsilon}\|)$ terms, we have:

$$\nabla_\theta \left(\left\{ L(\mathcal{D}, \hat{\theta}) + \epsilon \cdot \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\} + \Delta_{i,\epsilon} \cdot \nabla_\theta \left\{ L(\mathcal{D}, \hat{\theta}) + \epsilon \cdot \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\} \right) = 0. \tag{13}$$

Since $\hat{\theta}$ is a indeed a minimizer of the loss function $\ell(.)$, then we have $\nabla_\theta \ell(.) = 0$. Thus, (13) reduces to the following condition:

$$\left\{ \epsilon \cdot \nabla_\theta \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\} + \Delta_{i,\epsilon} \cdot \left\{ \nabla_\theta^2 \ L(\mathcal{D}, \hat{\theta}) + \epsilon \cdot \nabla_\theta^2 \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\} = 0. \tag{14}$$

By solving for ∇_θ, we have

$$\Delta_{i,\epsilon} = - \left\{ \nabla_\theta^2 \ L(\mathcal{D}, \hat{\theta}) + \epsilon \cdot \nabla_\theta^2 \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\}^{-1} \cdot \left\{ \epsilon \cdot \nabla_\theta \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\}, \tag{15}$$

which can be approximated by keeping only the $O(\epsilon)$ terms as follows:

$$\Delta_{i,\epsilon} = - \left\{ \nabla_\theta^2 \ L(\mathcal{D}, \hat{\theta}) \right\}^{-1} \cdot \left\{ \epsilon \cdot \nabla_\theta \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\}. \tag{16}$$

Noting that $\nabla_\theta^2 \ L(\mathcal{D}, \hat{\theta})$ is the Hessian matrix $H_{\hat{\theta}}$, we have:

$$\Delta_{i,\epsilon} = -H_{\hat{\theta}}^{-1} \cdot \left\{ \epsilon \cdot \nabla_\theta \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\}. \tag{17}$$

By taking the derivative with respect to ϵ, we arrive at the expression for first order influence functions:

$$\mathcal{I}^{(1)}(\boldsymbol{x}_i, y_i) = \frac{\Delta_{i,\epsilon}}{\epsilon}\Big|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \cdot \nabla_\theta \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})). \tag{18}$$

Now let us examine the second order influence functions. In order to obtain $\mathcal{I}^{(2)}(\boldsymbol{x}_i, y_i)$, we need to differentiate (14) after omitting the $O(\epsilon)$ once again as follows:

$$\left\{ 2\Delta_{i,\epsilon} \cdot \epsilon \cdot \nabla_\theta^2 \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\} + \left\{ \Delta_{i,\epsilon}^2 \cdot \nabla_\theta^2 \ L(\mathcal{D}, \hat{\theta}) + \Delta_{i,\epsilon} \cdot \Delta_{i,\epsilon} \cdot \nabla_\theta^3 \ L(\mathcal{D}, \hat{\theta}) \right\} = 0. \tag{19}$$

Where we have applied the chain rule to obtain the above. By substituting $\nabla_\theta^2 \ L(\mathcal{D}, \hat{\theta}) = H_\theta$ and dividing both sides of (19) by ϵ^2, we have

$$\left\{ 2 \, \frac{\Delta_{i,\epsilon}}{\epsilon} \cdot \nabla_\theta^2 \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right\} + \left\{ \frac{\Delta_{i,\epsilon}^2}{\epsilon^2} \cdot H_\theta + \left(\frac{\Delta_{i,\epsilon}}{\epsilon} \right)^2 \cdot \nabla_\theta^3 \ L(\mathcal{D}, \hat{\theta}) \right\} = 0. \tag{20}$$

Thus, by re-arranging (19), we can obtain $\mathcal{I}^{(2)}(\boldsymbol{x}_i, y_i)$ in terms of $\mathcal{I}^{(1)}(\boldsymbol{x}_i, y_i)$ as follows:

$$\mathcal{I}^{(2)}(\boldsymbol{x}_i, y_i) = -H_\theta^{-1} \left(\left(\mathcal{I}^{(1)}(\boldsymbol{x}_i, y_i) \right)^2 \cdot \nabla_\theta^3 \ L(\mathcal{D}, \hat{\theta}) + 2 \, \mathcal{I}^{(1)}(\boldsymbol{x}_i, y_i) \cdot \nabla_\theta^2 \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) \right).$$

Similarly, we can obtain the k^{th} order influence function, for any $k > 1$, by repeatedly differentiating equation (14) k times, i.e.,

$$\frac{\partial}{\partial \epsilon^k} \left\{ \epsilon \cdot \nabla_\theta \ \ell(y_i, f(\boldsymbol{x}_i; \hat{\theta})) + \Delta_{i,\epsilon} \cdot \nabla_\theta^2 \ L(\mathcal{D}, \hat{\theta}) \right\} = 0. \tag{21}$$

and solving for $\partial \Delta_{i,\epsilon}^k / \partial \epsilon^k$. By applying the higher-order chain rule to (21) (or equivalently, take the derivative of $\mathcal{I}^{(2)}(\boldsymbol{x}_i, y_i)$ for $k-2$ times), we recover the expressions in Definition 2 and Lemma 3 in (Giordano et al., 2019).

C. Theorem 1

Theorem 1 follows from Theorem 1 in (Barber et al., 2019) for $m \to \infty$ when all HOIFs exist.

Recall that the exact DJ interval width is bounded above by:

$$W(\widehat{\mathcal{C}}^{(\infty)}_{\alpha,n}(\boldsymbol{x};\hat{\theta})) \leq 2\,\widehat{Q}_{\alpha,n}(\mathcal{R}_n) + 2\,\widehat{Q}_{\alpha,n}(\mathcal{V}_n(\boldsymbol{x})). \tag{22}$$

Since the term $\widehat{Q}_{\alpha,n}(\mathcal{R}_n)$ is constant for any $\boldsymbol{x}$, discrimination boils down to the following condition:

$$\mathbb{E}[\,\widehat{Q}_{\alpha,n}(\mathcal{V}_n(\boldsymbol{x}))\,] \geq \mathbb{E}[\,\widehat{Q}_{\alpha,n}(\mathcal{V}_n(\boldsymbol{x}'))\,] \Leftrightarrow \mathbb{E}[\,\ell(y, f(\boldsymbol{x};\hat{\theta}))\,] \geq \mathbb{E}[\,\ell(y', f(\boldsymbol{x}';\hat{\theta}))\,]. \tag{23}$$

Note that to prove the above, it suffices to prove the following:

$$\mathbb{E}[\,v_i(\boldsymbol{x})\,] \geq \mathbb{E}[\,v_i(\boldsymbol{x}')\,] \Leftrightarrow \mathbb{E}[\,\ell(y, f(\boldsymbol{x};\hat{\theta}))\,] \geq \mathbb{E}[\,\ell(y', f(\boldsymbol{x}';\hat{\theta}))\,]. \tag{24}$$

If the model is stable (based on the definition in (Bousquet & Elisseeff, 2002)), then a classical result by (Devroye & Wagner, 1979) states that:

$$\mathbb{E}[\,|\ell(y, f(\boldsymbol{x};\hat{\theta})) - \ell_n(y, f(\boldsymbol{x};\hat{\theta}))|^2\,] \approx \mathbb{E}[\,|\ell(y, f(\boldsymbol{x};\hat{\theta})) - \ell(y, f(\boldsymbol{x};\hat{\theta}_{-i}))|^2\,] + Const., \tag{25}$$

as $n \to \infty$, where $\ell_n(.)$ is the empirical risk on the training sample, and the expectation above is taken over $y \,|\, \boldsymbol{x}$. From (25), we can see that an increase in the LOO risk $\ell(y, f(\boldsymbol{x};\hat{\theta}_{-i}))$ implies an increase in the empirical risk $\ell_n(y, f(\boldsymbol{x};\hat{\theta}))$, and vice versa. Thus, for any two feature points $\boldsymbol{x}$ and $\boldsymbol{x}'$, if $v(\boldsymbol{x})$ is greater than $v(\boldsymbol{x}')$, then on average, the empirical risk at $\boldsymbol{x}$ is greater than that at $\boldsymbol{x}'$.

D. Experimental Details

D.1. Implementation of Baselines

In what follows, we provide details for the implementation and hyper-parameter settings for all baseline methods involved in Section 5.

Probabilistic backpropagation (PBP). We implemented the PBP method proposed in ((Hernández-Lobato & Adams, 2015)) with inference via expectation propagation using the `theano` code provided by the authors in (`github.com/HIPS/Probabilistic-Backpropagation`). Training was conducted via 1000 epochs.

Monte Carlo Dropout (MCDP). We implemented a `Pytorch` version of the MCDP method proposed in ((Gal & Ghahramani, 2016)). In all experiments, we tuned the dropout probability using Bayesian optimization to optimize the AUC-ROC performance on the training sample. We used 1000 samples at inference time to compute the mean and variance of the predictions. The credible intervals were constructed as the $(1-\alpha)$ quantile function of a posterior Gaussian distribution defined by the predicted mean and variance estimated through the Monte Carlo outputs. Similar to the other baselines, we conducted training via 1000 epochs for the SGD algorithm.

Bayesian neural networks (BNN). We implemented BNNs with inference via stochastic gradient Langevin dynamics (SGLD) ((Welling & Teh, 2011)). We initialized the prior weights and biases through a uniform distribution over $[-0.01, 0.01]$. We run 1000 epochs of the SGLD inference procedure and collect the posterior distributions to construct the credible intervals.

Deep ensembles (DE). We implemented a `Pytorch` version of the DE metho (without adversarial training)d proposed in ((Lakshminarayanan et al., 2017)). We used the number of ensemble members $M = 5$ as recommended in the recent study in ((Ovadia et al., 2019)). Predictions of the different ensembles were averaged and the confidence interval was estimated as 1.645 multiplied by the empirical variance for a target coverage of 90%. We trained the model through 1000 epochs.

References

Barber, R. F., Candes, E. J., Ramdas, A., and Tibshirani, R. J. Predictive inference with the jackknife+. *arXiv preprint arXiv:1905.02928*, 2019.

Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Devroye, L. and Wagner, T. Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions on Information Theory*, 25(2):202–207, 1979.

Fernholz, L. T. *Von Mises calculus for statistical functionals*, volume 19. Springer Science & Business Media, 2012.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*, pp. 1050–1059, 2016.

Giordano, R., Jordan, M. I., and Broderick, T. A higher-order swiss army infinitesimal jackknife. *arXiv preprint arXiv:1907.12116*, 2019.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. *Robust statistics: the approach based on influence functions*, volume 196. John Wiley & Sons, 2011.

Hernández-Lobato, J. M. and Adams, R. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning (ICML)*, pp. 1861–1869, 2015.

Huber, P. J. and Ronchetti, E. M. Robust statistics john wiley & sons. *New York*, 1(1), 1981.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6402–6413, 2017.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.