Understanding Black-box Predictions via Influence Functions

Pang Wei Koh, Percy Liang

Presenter: Wei-Cheng Lee, Cheng-Wei Tsai
Advisor: Shou-De Lin

Outline

Motivation

- **Why:**
  We often ask "Why did the system make this predictions?" We want the model that are not only high-performing but also explainable.
- **Then:**
  By understanding the model, we hope that we can improve the model (Amershi et al.,2015) discover new science(Shrikumar et al., 2016) and provide end-users with explanations of actions that impact them (Goodman Flaxman,2016).
- **Challenge:**
  The best-performing models in many domains such as deep neural networks for image and speech recognition are complicated, black-box models whose predictions seem hard to explain.

- **Previous Focus:** Why a fixed model leads to particular predictions
  - Locally fitting a simpler model around the test point (Why should I trust you, Riberiro et al., 2016)
  - Perturbing the test point to see change in predictions (Simonyan et al., 2013; Li et al., 2016b; Datta et al., 2016; Adler et al., 2016)
- **This Paper:** Try to formalize the impact of a training point on predictions.
  - Why would happen to the model if we don't have a training point.
  - Use statistical tool, influence function, to tackle with this problem.

Outline

Notations

- **Empirical Risk Minimization**
  Consider a prediction from input space $\mathcal{X}$(images) to $\mathcal{Y}$(labels). Given training points $\{z_1, z_2, ..., z_n | z_i = (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$. For a point z and $\theta \in \Theta$, let $L(z, \theta)$ be the loss. The empirical risk minimizer $\hat{\theta} \stackrel{def}{=} \text{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta)$.
- **Removing a training point**
  We want to study the change in model parameters due to removing z. That is $\hat{\theta}_{-z} - \hat{\theta}$. ($\hat{\theta}_{-z} \stackrel{def}{=} \text{argmin}_{\theta \in \Theta} \frac{1}{n} \sum_{z_i \neq z} L(z_i, \theta)$)
- **How to calculate?**
  If we leave one out the entire training set and re-train the model, it will be prohibitively slow. We will use influence function to approximate it and show it is a good approximation.

Influence Function-Upweighing a training point

- The idea is to compute the parameter change if z were upweighted by small ϵ given us new parameters

$$\hat{\theta}_{\epsilon,z} \stackrel{def}{=} \underset{\theta \in \Theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$$

- A classical result (Cook & Weisberg, 1982) shows when ϵ approaches zero

$$\mathcal{I}_{up,params(z)} \stackrel{def}{=} \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\Big|_{\epsilon=0} = \underbrace{-H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta})}_{\color{red}{Take\ \ a\ \ Newton\ \ step}}$$

where $H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta}^{2} L(z_i, \hat{\theta})$. We can linear approximate the parameter change due to removing z by

$$\hat{\theta}_{-z} - \hat{\theta} = \hat{\theta}_{-\frac{1}{n},z} - \hat{\theta}_{0,z} \approx -\frac{1}{n} \mathcal{I}_{up,params(z)}$$

Influence Function-Upweighing a training point

- Our goal is to estimate the influence of **removing a point z** on the loss at a test point z_{test} which has a closed form using chain-rule:

$$\mathcal{I}_{up,loss(z,z_{test})} \overset{def}{=} \frac{dL(z_{test}, \hat{\theta}_{\epsilon,z})}{d\epsilon} = \nabla_\theta L(z_{test}, \hat{\theta})^\top \frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}|_{\epsilon=0}$$
$$= -\nabla_\theta L(z_{test}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})$$

Influence Function-Perturbing a training point

- We want to further ask what would the model's prediction change if **a training input were modified**. $z \to z_\delta \stackrel{def}{=} (x+\delta, y)$
- Consider $\hat{\theta}_{\epsilon, z_\delta, -z} \stackrel{def}{=} \text{argmin}_{\theta \in \Theta} \frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z_\delta, \theta) - \epsilon L(z, \theta)$ we have

$$\begin{aligned} \frac{d\hat{\theta}_{\epsilon, z_\delta, -z}}{d\epsilon}\Big|_{\epsilon=0} &= \mathcal{I}_{up,params(z_\delta)} - \mathcal{I}_{up,params(z)} \\ &= -H_{\hat{\theta}}^{-1}(\nabla_\theta L(z_\delta, \hat{\theta}) - \nabla_\theta L(z, \hat{\theta})) \end{aligned}$$

- So we can linear approximate

$$\hat{\theta}_{z_\delta, -z} - \hat{\theta} \approx -\frac{1}{n}(\mathcal{I}_{up,params(z_\delta)} - \mathcal{I}_{up,params(z)})$$

which gives us a closed form estimate of the effect $z \mapsto z_\delta$ on the model.

Influence Function-Perturbing a training point

- Assume $\mathcal{X} \subseteq \mathbb{R}^d$ continuous and parameter space $\Theta \subseteq \mathbb{R}^p$ and L is differentiable in θ and x. As $||\delta|| \to 0$,

$$\nabla_\theta L(z_\delta, \hat{\theta}) - \nabla_\theta L(z, \hat{\theta}) \approx [\nabla_x \nabla_\delta L(z, \hat{\theta})]\delta$$

  We thus have $\hat{\theta}_{z_\delta,-z} - \hat{\theta} \approx -\frac{1}{n} H_{\hat{\theta}}^{-1} [\nabla_x \nabla_\delta L(z, \hat{\theta})]\delta$
- Using chain-rule we can approximate the effect $z \mapsto z_\delta$ has on the loss at z_{test}

$$\begin{aligned}
\mathcal{I}_{pert,loss}(z, z_{test})^\top &\stackrel{def}{=} \frac{dL(z_{test}, \hat{\theta}_{,z_\delta,-z})}{d\delta}|_{\delta=0} \\
&= \nabla_\theta L(z_{test}, \hat{\theta})^\top \frac{d\hat{\theta}_{z_\delta,-z}}{d\delta}|_{\delta=0} \\
&= -\nabla_\theta L(z_{test}, \hat{\theta})^\top H_{\hat{\theta}}^{-1} \nabla_x \nabla_\delta L(z, \hat{\theta})
\end{aligned}$$

- $\mathcal{I}_{pert,loss}(z, z_{test})^\top \delta$ measures the perturbing effect of $z \mapsto z_\delta$ which can be used to construct training-set attacks.

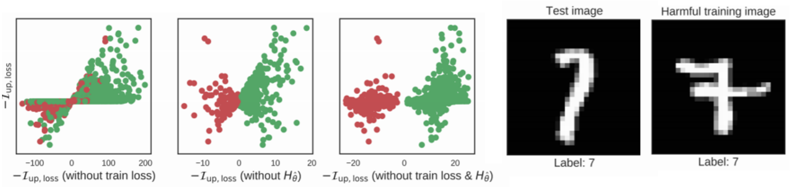
Relation to Euclidean distance

- We can use Euclidean distance to find the training points most related to a test point.(Why should I trust you, Riberiro et al., 2016)
- If all the points have the same norm, it is equivalent to finding $argmin_{x \in \{training\}} x.x_{test}$. We compare it to $\mathcal{I}_{up,loss}(z, z_{test})$ on a logistic regression model.
- $p(y|x) = \sigma(y\theta^\top x)$,$y \in \{-1, +1\}$ and $\sigma(t) = \frac{1}{1+exp(-t)}$. The loss is given by $L(z, \theta) = log(1 + exp(-y\theta^\top x))$. We will have $\mathcal{I}_{up,loss}(z, z_{test})$:

$$-y_{test} y \cdot \sigma(-y_{test}\theta^\top x_{test}) \cdot \sigma(-y\theta^\top x) \cdot x_{test}^\top H_{\hat{\theta}}^{-1} x.$$

Relation to Euclidean distance

- Green dots are train images as the test image while red dots are 1's.



$$\mathcal{I}_{up,loss}(z, z_{test}) = -y_{test} y \cdot \sigma(-y_{test}\theta^\top x_{test}) \cdot \underbrace{\sigma(-y\theta^\top x)}_{\propto training \ \ loss} \cdot x_{test}^\top H_{\hat{\theta}}^{-1} x.$$

1. Without loss, we over estimate the influence of many training points.
2. Without Hessian $x_{test}^\top x \geq 0$ always, $sign(\mathcal{I}) = -y_{test} y$, but we may have harmful image having same label as y_{test}.
3. Without both, it is scaled $x_{test}^\top x$ fails to lies on diagonal to accurately capture influence.

Outline

Outline