Class-based Influence Functions for Error Detection

Nguyen Duc-Thang*, Hoang Thanh-Tung*, Quan Tran*, Huu-Tien Dang, Nguyen Ngoc-Hieu, Anh Dau, Nghi Bui

FPT Software AI Center, Adobe Research

Contributions

We propose class-based influence function algorithms for automatic error detection in large-scale datasets. **Code**: `https://github.com/Fsoft-AIC/Class-Based-Influence-Functions`

Main contributions

- We provide empirical and theoretical reasons for the instability of Influence Functions (IFs).
- We propose IFs-class, variants of IFs that use class information to enhance stability.
- Extensive experiments show that IFs-class significantly improves the performance and stability of IFs while incurring no additional computational cost.

Abstract

Influence functions (IFs) are a powerful tool for detecting anomalous examples in large-scale datasets. However, they are unstable when applied to deep networks. In this paper, we provide an explanation for the instability of IFs and develop a solution to this problem. We show that IFs are unreliable when the two data points belong to two different classes. Our solution leverages class information to improve the stability of IFs. Extensive experiments show that our modification significantly improves the performance and stability of IFs while incurring no additional computational cost.

Motivation

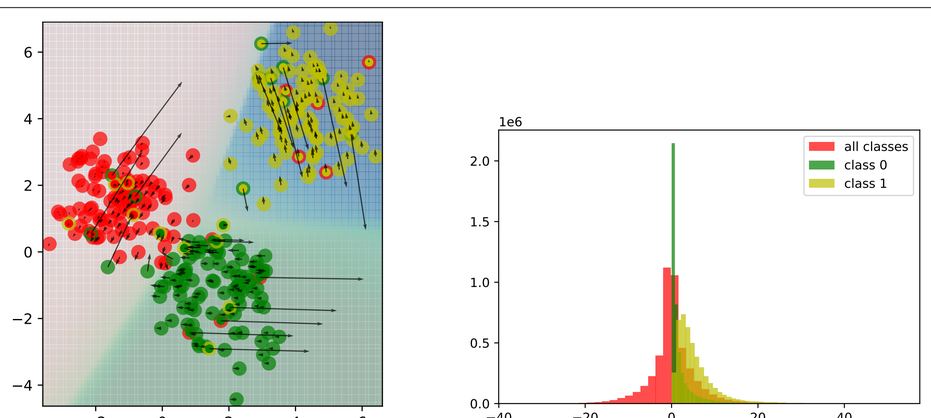

Figure 1. (Left) Gradients of erroneous data are opposite to gradients of clean data from the true class. (Right) Gradient dot products of data points from two classes are normally distributed with a sharp peak at 0. The gradient dot product of data points from the same class is almost always positive.

Observations

- Erroneous data points have a very negative influence on data points from the true class but have a small and noisy influence on data points from other classes
- Clean data points almost always have a positive influence on data points from the same class.

Proposed method: IFs-class

Algorithm 1 Class based influence function for error detection.

Require:

$\mathcal{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^{n}$: a big noisy dataset
C: number of classes
$\mathcal{Z}'_k = \{\mathbf{z}'^{(j_k)}\}_{j_k=1}^{m_k}$: clean data from class k
$\mathcal{Z}' = \bigcup_{k=1}^{C} \mathcal{Z}'_k$: a clean reference dataset
$f_{\hat{\boldsymbol{\theta}}}$: a deep model pretrained on $\mathcal{Z}$
$\text{sim}(\cdot,\cdot)$: a similarity measure in Tab. 3

Ensure: $\hat{\mathcal{Z}}$: data points in $\mathcal{Z}$ ranked by score

for $\mathbf{z}^{(i)} \in \mathcal{Z}$ **do**
  for $k = 1, ..., C$ **do**
    $s_k^{(i)} = \frac{\sum_{j=1}^{m_k} \text{sim}(\nabla_{\hat{\boldsymbol{\theta}}} \ell(\mathbf{z}^{(i)}), \nabla_{\hat{\boldsymbol{\theta}}} \ell(\mathbf{z}'^{(j_k)}))}{m_k}$
  end for
  $s^{(i)} = \min_k(s_k^{(i)})$
end for
$\hat{\mathcal{Z}} = \text{sort}(\mathcal{Z}, \text{key} = \boldsymbol{s}, \text{ascending} = \text{True})$
return $\hat{\mathcal{Z}}$

- Unlike the original IFs, IFs-class are unaffected by the noise from other classes and thus, have lower variances.
- IF-class has the same computational complexity as the IFs-based error detection algorithm.

Experiments

Experiment Settings

Tasks and Datasets

- Text classification: IMDB, SNLI, and BigCloneBench datasets
- Named Entity Recognition: CoNLL2003 dataset

Models

- BERT
- CodeBERT

Evaluation

- Inject $p\%$ error in each of the datasets.
- Select top $q\%$ most harmful data points from the sorted dataset $\hat{\mathcal{Z}}$ and check how many percent of the selected data points are really erroneous.

Results

- Our class-based influence score significantly improves the error detection performance and reduces the variance.
- When the reference set is noisy, IF-class algorithms are much more robust and their performance decreases only slightly.

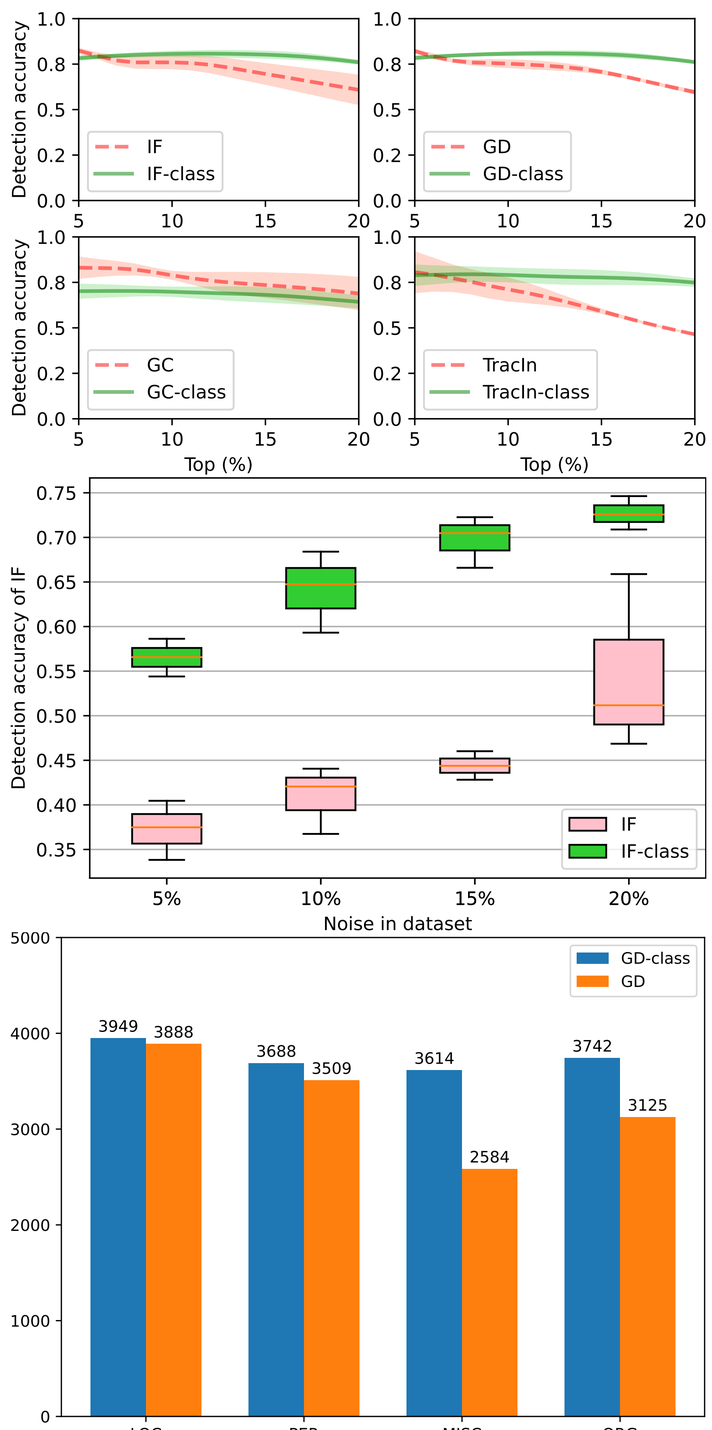

Figure 2. (top) Error detection accuracy on SNLI dataset with $p = 20\%$. (middle) Error detection accuracy of IF and IF-class on IMDB dataset with different values of p. (bottom) The number of erroneous NER tokens detected by GD and GD-class at $p = 30\%, r = 30\%, q = 9\%$, grouped by entity types of the erroneous tokens.

Table 1. The result of GD and GD-class on SNLI dataset when the reference set is a random (noisy) subset of the training set.

Method	top 5%	top 10%	top 15%	top 20%
GD	75.33	59.37	43.87	34.49
GD-Class	73.85	70.84	67.28	64.29