Identifying influential points to explain predictions

Arthur MAILLART
Laboratoire SAF

Objectives

In this work, we try to get insight on which points are important for a Machine Learning method to predict an instance that the model didn't see before. From the work of Pang and Percy [2] we implement a method based on influence functions for tabular data. The two main objectives are :

- Check and reproduce the results of the article and confirm that the method works on logistic regression (explainable model)
- Extract relevant information from the data to explain the decision of the method

Introduction

Complex Machine Learning models have taken their place into Data Lab of many insurance companies. Some of them have proven their interest on insurance data. Unfortunately, it is rather difficult to adopt a predictive model that you can not explain simply. That is why there is a growing interest in intelligibility in Machine Learning.

As proposed by Pang and Percy [2], the influence for a prediction at test point is defined by the variation of loss at test point between two models : the first trained on the whole training set and the second trained on the training set from which one removes a point. To provide an explanation, the authors extracted the most influential points and display images (visual artifacts). In this work, we adapt the method to tabular data and try to transform the information about the points influences into understandable indicators or explanations.

We show that the method based on influence functions can be useful to get information from the model. For example, we are able to identify points that can be considered as outliers by a given model.

Methodology

We want to approximate the loss variation caused by the deletion of a point in the training set to understand its effect on a model. Starting from the definition of influence function (from robust statistics theory [1]), Pang and Percy [2] show that upweighting a point by ϵ lead to a good approximation of this quantity.

For a given Machine Learning model, let

- $z_i = (x_i, y_i),\ 1, ..., n$ be the training set points,
- $L(z_i, \theta)$ the loss of the model evaluated in z_i,
- $\theta^* \overset{def}{=} \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta)$ the vector of optimal parameters,
- $\theta^*_{\epsilon,z}$ the optimal parameters after upweighting point z by ϵ,
- $H_{\theta^*} = \frac{1}{n}\sum_{i=1}^{n} \nabla^2_\theta L(z_i, \theta^*)$ the hessian of the empirical risk

The article above define three quantities of interest

$$\mathcal{I}_{up,params}(z) \overset{def}{=} \left.\frac{d\theta^*_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0} = -H^{-1}_{\theta^*}\nabla_\theta L(z, \theta^*)$$

which measures the change in parameters if a training point is upweighted by ϵ. Setting $\epsilon = -\frac{1}{n}$ we can approximate the effect of removing a point from the training set on the parameters.

$$\begin{aligned}\mathcal{I}_{up,loss}(z, z_{test}) &\overset{def}{=} \left.\frac{dL(z_{test}, \theta^*_{\epsilon,z})}{d\epsilon}\right|_{\epsilon=0} \\ &= -\nabla_\theta L(z_{test}, \theta^*)^T H^{-1}_{\theta^*}\nabla_\theta L(z, \theta^*)\end{aligned}$$

It provides for a given test point the influence of each train point for this prediction. This is really informative because we can learn which points are helpful and harmful to the model and to what extent.

$$\begin{aligned}\mathcal{I}_{pert,loss}(z, z_{test}) &\overset{def}{=} \nabla_\delta L(z_{test}, \theta^*_{z\delta}, -z)|_{\delta=0} \\ &= -\nabla_\theta L(z_{test}, \theta^*)^T H^{-1}_{\theta^*}\nabla_x\nabla_\theta L(z, \theta^*)\end{aligned}$$

where $z_\delta \overset{def}{=} (x+\delta, y)$. In other words, this quantifies the change in loss as we modify the value of a train point z.
For the moment we have implemented and tested the two first indicators, and we will investigate the last one soon.

Experiments with logistic regression

We have implemented the method for tabular data and simulated a data set with two dimensions and two classes. This simple example is very informative about the results given by the method.

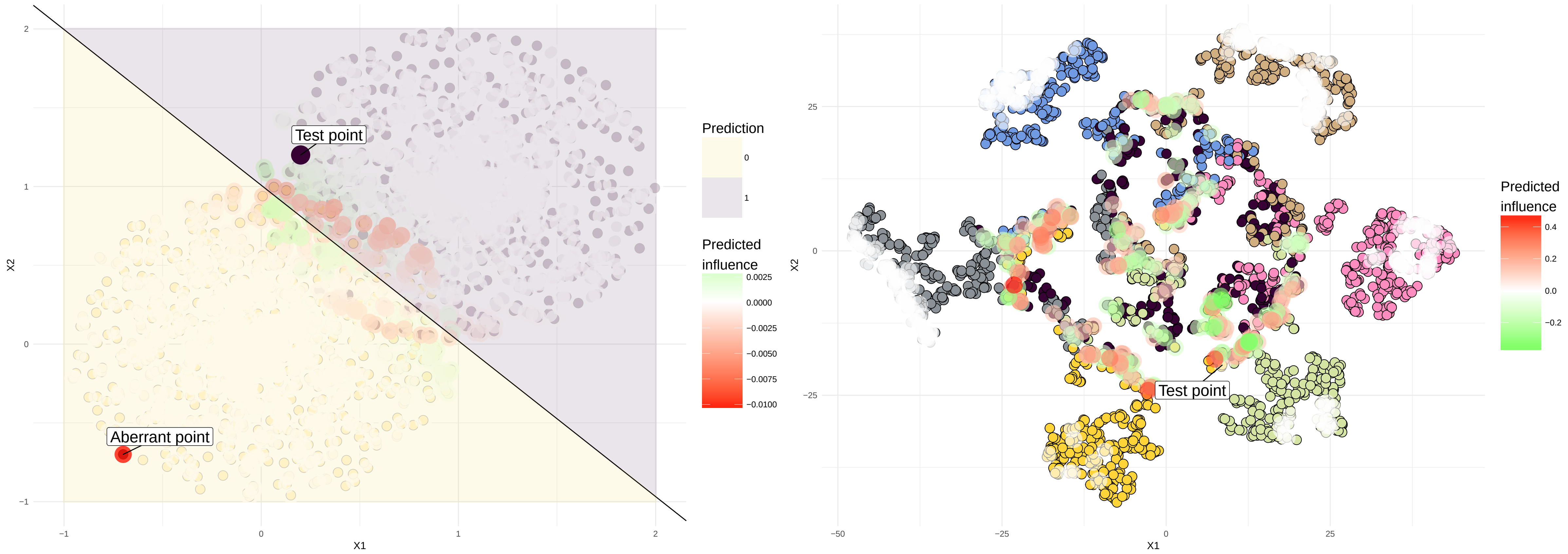

Figure 1: Left - Influential points for binary logistic regression (data set = two overlapping disks) – Right - 2D-visualisation of influential points for multi-class logistic regression (data set = seven overlapping cubes) (harmful points/helpful points)

As we can see above, the algorithm has identified the points nearby the decision boundary as influential to predict the test point. That means that the logistic regression mostly needs points that help to adjust the decision boudary to predict. Moreover, we can see that the method has found an aberrant value for the logistic regression.
To confirm our first intuitions, we have simulated a data set in three dimensions. It consists of seven cubes (seven classes) that overlap partially. To represent this data set we used t-SNE [3] to create a 2D-visualisation.

Conclusion

Influence functions in Machine Learning show promising results. So far we have confirmed on logistic regression (binary/multi-class) that the method can help users determine influential points and doing this get an insight about how a given model work. Furthermore, this method can identify aberrant values for a given model.
For future work we have planned to implement the last indicator to extract information about influential points.

[1] Peter J. Huber.
Robust Statistics.
Wiley series in probability and mathematical statistics. Wiley, 1981.

[2] Pang Wei Koh and Percy Liang.
Understanding black-box predictions via influence functions.
In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1885–1894, 2017.

[3] Laurens van der Maaten and Geoffrey Hinton.
Visualizing data using t-SNE.
Journal of Machine Learning Research, 9:2579–2605, 2008.