Fast Conformal Classification using Influence Functions

Umang Bhatt USB20@CAM.AC.UK
University of Cambridge, Cambridge, United Kingdom
Adrian Weller AW665@CAM.AC.UK
University of Cambridge, Cambridge, United Kingdom
The Alan Turing Institute, London, United Kingdom
Giovanni Cherubin GCHERUBIN@TURING.AC.UK
The Alan Turing Institute, London, United Kingdom

Editor: Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin and Khuong An Nguyen

Abstract

We use influence functions from robust statistics to speed up full conformal prediction. Traditionally, conformal prediction requires retraining multiple leave-one-out classifiers to calculate p-values for each test point. By using influence functions, we are able to approximate this procedure and to speed up considerably the time complexity.

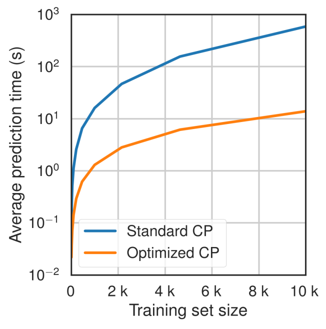
Keywords: Conformal prediction, influence functions, optimization.

1. Introduction

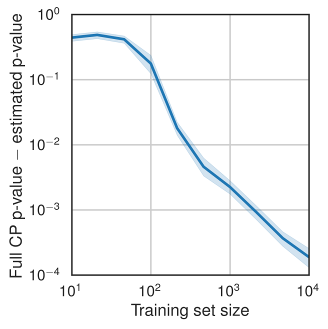
Background Conformal prediction (CP) is a post-hoc approach to providing guarantees on the outcomes of machine learning models. Full CP (or "transductive CP") is notoriously computationally expensive, albeit generally having the best statistical power among CP-derived methods, such as split CP (Vovk et al., 2005), cross-CP (Vovk, 2015), and the jackknife+ (Barber et al., 2021). Recent work optimized "deleted" full CP for the family of non-conformity measures that support incremental and decremental learning, by speeding up the leave-one-out (LOO) procedure required for the p-value calculation (Cherubin et al., 2021). However, this approach may not scale to more complex machine learning models like neural networks. In this work, we propose the use of influence functions from robust statistics (Hampel, 1974) to approximate the result of the LOO procedure and therefore to speed up deleted full CP for more general machine learning models.

Non-conformity Measure Consider a model parametrized by $\theta_{\mathcal{D}} \in \Theta$ trained on data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$; we assume that $\theta_{\mathcal{D}} = \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i\in[n]} l(x_i, y_i; \theta)$ is the empirical risk minimizer, where l is a twice-differentiable loss $l(x, y; \theta)$ that is convex in θ. This setup permits simple models like logistic regression and complex models like neural networks. We define a non-conformity measure for model $\theta_{\mathcal{D}}$ as its loss at a test point, $A((x, y); \mathcal{D}) = l(x, y; \theta_{\mathcal{D}})$.

Speeding up CP via Influence Functions Running CP to compute a p-value for a test object x and a candidate label $\hat{y}$ requires recomputing the nonconformity measure (and, therefore, retraining model θ) $n+1$ times. Our optimization idea is to first learn a model $\theta_{\mathcal{Z}}$ on an augmented dataset $\mathcal{Z} = \mathcal{D} \cup \{(x, \hat{y})\}$, and then to use influence functions to approximate the non-conformity scores that would be obtained in the LOO procedure upon dropping training points (Algorithm 1). Note that $A((x, y); \mathcal{Z})$ depends on a model learned on $\mathcal{Z}$. In full CP, we calculate $A((x,y);\mathcal{Z}\setminus\{(x,y)\})$ for each $(x,y)\in\mathcal{Z}$: this requires learning $n+1$ models. Koh and Liang (2017) perform a first-order Taylor expansion to approximate the loss of a model $\theta_{\mathcal{Z}}$ at (x,y) upon dropping (x,y) from $\mathcal{Z}$ as $I_{\mathcal{Z}}(x,y) := -\nabla_\theta\, l(x,y;\theta_{\mathcal{Z}})^\intercal H_{\theta_{\mathcal{Z}}}^{-1}\nabla_\theta\, l(x,y;\theta_{\mathcal{Z}})$, where $H_{\theta_{\mathcal{Z}}} = \frac{1}{|\mathcal{Z}|}\sum_{(x_i,y_i)\in\mathcal{Z}}\nabla_\theta^2 l(x_i,y_i;\theta)$ is the Hessian of the loss l. As such, we approximate A for a test point (x,y) as $\alpha = A((x,y);\mathcal{Z}) - I_{\mathcal{Z}}(x,y)$, where $A((x,y);\mathcal{Z})$ represents the loss at a test point $(x,\hat{y})$ for a model trained on the full augmented dataset $\mathcal{Z}$. Instead of training $n+1$ models for full CP, our approximation only requires training one model on $\mathcal{Z}$.



(a) Time



(b) Validity

Algorithm 1 CP+IF (Ours)

```
input $x,\ \hat{y},\ A,\ \mathcal{D}=\{(x_i,y_i)\}_{i=1}^n$
1: $\mathcal{Z}=\mathcal{D}\cup\{(x,\hat{y})\}$
2: Learn $\theta_{\mathcal{Z}}$ via ERM
3: Calculate Hessian inverse, $H_{\theta_{\mathcal{Z}}}^{-1}$
4: for $i\in[1,n+1]$ do
5:     $g=\nabla_\theta\, l(x_i,y_i;\theta_{\mathcal{Z}})$
6:     $I_{\mathcal{Z}}(x_i,y_i)=-g^\intercal H_{\theta_{\mathcal{Z}}}^{-1}g$
7:     $\alpha_i=A((x_i,y_i);\mathcal{Z})-I_{\mathcal{Z}}(x_i,y_i)$
8: end for
9: $p_{(x,\hat{y})}=\frac{\#\{i=1,\ldots,n\,:\,\alpha_i\geq\alpha_{n+1}\}+1}{n+1}$
output $p_{(x,\hat{y})}$
```

2. Preliminary Results

We generate synthetic 30-dimensional data points from two normally-distributed clusters for a binary classification problem by using `scikit-learn`'s `make_classification()` function (Pedregosa et al., 2011). We evaluate our method for logistic regression models. For a test set of 100 points, we measure the average time required to make a test prediction for training sets of various sizes. Fig 1(a) shows that our approach is considerably more time-efficient than full CP. In Fig 1(b), we evaluate the quality of our predictions as the absolute difference between the p-values obtained via full CP and our method. We observe a large difference for small training sets; however, in that case, full CP is feasible to run. As the training set gets larger, our method works better: the average difference has order 10^{-3}. We observe a similar behavior when comparing the p-values' fuzziness (Vovk et al., 2016). For 100 training examples, full CP has average fuzziness 0.079, our method has 0.36. For 1k examples, their respective fuzziness are 0.0269 and 0.0278. For 10k, they are both 0.0258.

3. Future Work

We envision the use of influence functions has a great potential in the context of CP. In this proposal, we are simply doing decremental learning (also known as *machine unlearning*) to approximate our p-value; indeed, our approach requires calculating H_θ^{-1} for each test point added to $\mathcal{D}$. Instead, we can consider a string of n rank-k corrections to our original Hessian inverse using tools like the Woodbury matrix identity (Woodbury, 1950; Sherman and Morrison, 1950). This means we would only need a single Hessian inversion to calculate p-values for all test points. Moreover, we would like to bound the error of our approximation, which requires a careful study of the underestimation of influence functions (Koh et al., 2019). We then can characterize the effect of our approximation on the p-values we obtain. Developing approximations to CP will enable a form of full CP that is fast and scalable.

References

Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

Giovanni Cherubin, Konstantinos Chatzikokolakis, and Martin Jaggi. Exact optimization of conformal predictors via incremental and decremental learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1836–1845. PMLR, 18–24 Jul 2021.

Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

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 (ICML 2017)*, pages 1885–1894. Journal of Machine Learning Research, 2017.

Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems*, pages 5254–5264, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jack Sherman and Winifred J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Statist.*, 21(1):124–127, 1950. doi: 10.1214/aoms/1177729893.

Vladimir Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74(1):9–28, 2015.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alexander Gammerman. Criteria of efficiency for conformal prediction. In *Symposium on conformal and probabilistic prediction with applications*, pages 23–39. Springer, 2016.

Max A Woodbury. Inverting modified matrices. Statistical Research Group, Memo. Rep. 42, Princeton University, Princeton, N. J, 1950.