Detecting Covariate Drift with Explanations

Steffen Castle$^{1[0000-0002-4538-9270]}$,
Robert Schwarzenberg$^{1[0000-0002-8819-1635]}$, and
Mohsen Pourvali$^{2[0000-0003-2653-9613]}$

[1] German Research Center for Artificial Intelligence (DFKI)
first.last@dfki.de
[2] Lenovo Research
mpourvali@lenovo.com

Abstract. Detecting when there is a domain drift between training and inference data is important for any model evaluated on data collected in real time. Many current data drift detection methods only utilize input features to detect domain drift. While effective, these methods disregard the model's evaluation of the data, which may be a significant source of information about the data domain. We propose to use information from the model in the form of explanations, specifically *gradient times input*, in order to utilize this information. Following the framework of Rabanser et al. [11], we combine these explanations with two-sample tests in order to detect a shift in distribution between training and evaluation data. Promising initial experiments show that explanations provide useful information for detecting shift, which potentially improves upon the current state-of-the-art.

Keywords: Data drift · XAI · two-sample tests

1 Introduction

Many AI models are trained on data that is gathered during an initial collection period and evaluated on data that is collected in real time. Real-time data is often subject to drift due to changes in data collection methodology, sampling differences, or a drift in underlying variables over time. Such drift can be problematic and may result in a degradation in the performance of the model. Therefore, detecting drift is important to ensure that a model performs optimally.

Covariate drift, which is a drift in the input domain, is commonly detected by comparing inputs from the training domain to the test domain using statistical tests. However, using only information from the inputs disregards another significant source of information about the data: The model itself, which is trained in the input domain.

One way to represent this information from the model is with explanations. Explanations are generated for a model to provide insight into a model's evaluation of an input. One type of explanation, attribution, denotes the importance of each input feature to the model's output on a data point. Many attribution techniques use model gradients as feature importance values, or more specifically, the gradient of the class output neuron with respect to the input feature [14]. Gradient times input [13] simply multiplies the attribution provided by the gradient by the input itself.

In order to use information from the model to detect drift, we introduce a model-based drift detection method in the framework of Rabanser et al. [11] that employs attributions in the form of gradient times input to detect data drift.

1.1 Related Work

Much work has already been done on detecting data drift. The main source of inspiration for our work, Rabanser et al. [11] employs two-sample tests, which compare datasets as samples from probability distributions in order to identify any divergence. Distribution-free tests are used to compare the distance between distributions and, based on this distance, determine the probability that the samples come from the same distribution. These tests are applied to representations of the input consisting of various dimensionality reductions.

Our method is not the first to use the model's representation of the data in detecting drift. Other methods, such as Black Box Shift Detection (BBSD)[9], make use of model output. BBSD was originally defined to detect prior-probability drift, but has also been applied to detect covariate drift [11]. Other methods use additional information from the model such as Elsahar et al. [2], where model confidence and reverse classification accuracy are used to detect drift conditioned on the model.

2 Methods

Our goal is to detect whether newly collected data has drifted compared to the initial training dataset. Formally, we compare samples from two distributions $X = \{x_1, x_2, ..., x_n\} \sim P_{train}(x)$ and $X' = \{x'_1, x'_2, ..., x'_n\} \sim P_{test}(x)$ to test the null hypothesis H_0 that the two samples come from the same distribution.

To accomplish this, we employ the MMD test [4] as employed by Rabanser et al. [11]. The MMD statistic represents the squared distance between the embedding means of distributions, that is:

$$\mathrm{MMD}(\mathcal{F}, p, q) = \|\mu_{\mathbf{P_{train}}} - \mu_{\mathbf{P_{test}}}\|^2_{\mathcal{F}} \tag{1}$$

where $\mu_{p_{train}}$ and $\mu_{p_{test}}$ are the mean embeddings of the distributions P_{train} and P_{test} in a reproducing kernel Hilbert space $\mathcal{F}$. The MMD test is distribution-free, meaning that it does not require any prior knowledge of the distribution types of P_{train} and P_{test}. The MMD statistic on X and X' should be large when P_{train} and P_{test} are different; the MMD kernel matrix can also be used to calculate a p-value for H_0 using a permutation test. A threshold α is chosen such that if the p-value $\leq \alpha$, H_0 can be rejected. We choose the standard $\alpha = 0.05$.

Our contribution to this framework is to introduce attributions to the two-sample drift detection procedure. We perform two-sample tests on representations of the input as in Rabanser et al. [11]. However, instead of dimensionality reduction, we substitute the original inputs with attribution maps consisting of the gradient times input. The attribution map $\phi(x)$ for a data point x and model output $f(x)$ is defined for gradient times input as

$$\phi(x) = \frac{\partial f(x)}{\partial x} \cdot x \tag{2}$$

3 Experiments[‡]

Following the framework of Rabanser et al. [11], shifts are artificially induced in the inputs of the test set and representations are produced, which are then compared to the original validation set with two-sample tests. In the image domain, we test the two main harmful types of drift identified by Rabanser et al. [11]: An image shift consisting of a random image translation of 5% or less, rotation of 10 degrees or less, and scaling of 10% or less (denoted small image shift in [11]), along with an adversarial shift, which replaces the test set with adversarial samples from FGSM [3]. For gradient times input, the model used to generate the explanations is a ResNet-50 model [5] trained on the train set of MNIST [8] with early stopping.

Input representations compared are those outlined by Rabanser et al. [11] plus the gradient times input methods. These representations consist of:

- No reduction (**NoRed**): A simple baseline consisting of original, unmodified inputs.
- Principal Components Analysis (**PCA**), Sparse Random Projection (**SRP**): Standard dimensionality reduction techniques detailed in [11]. As in the original paper, the input dimensionality is reduced to a size of 32.
- Autoencoder, Trained (**TAE**) and Untrained (**UAE**): The latent space of an autoencoder which has either been trained on the input domain, or has not received any training.
- Black Box Shift Detection, Softmax (**BBSDs**): The softmax-layer output of a model trained for classification on the training set.
- Gradient times input (**GradxInput**): The attribution produced by the gradient of the output neuron with highest activation with respect to the input multiplied by the input as in Equation 2.

We aim to detect drift with high sensitivity, that is, with as low a number of samples as possible. Thus we compare results from different methods with varying random sample sizes $s \in \{10, 20, 50, 100, 200, 500, 1000\}$. Each test is performed 15 times and the mean p-value is then determined.

We also evaluate on a simple sentiment classification task consisting of a DistilBERT [12] model pretrained on the Large Movie Review Dataset (IMDB)

[‡] Code available at `https://github.com/DFKI-NLP/xai-shift-detection`

[10] of movie reviews labeled as either positive or negative. In this case, the shift consists of adversarial reviews provided by TextFooler [6]. Representations are compared at the embedding level, with the gradient times input calculated with respect to the embedding. For this experiment, the autoencoder representations were not evaluated. Additionally, since only 1000 adversarial examples were available, $s = 1000$ was not tested for this task.

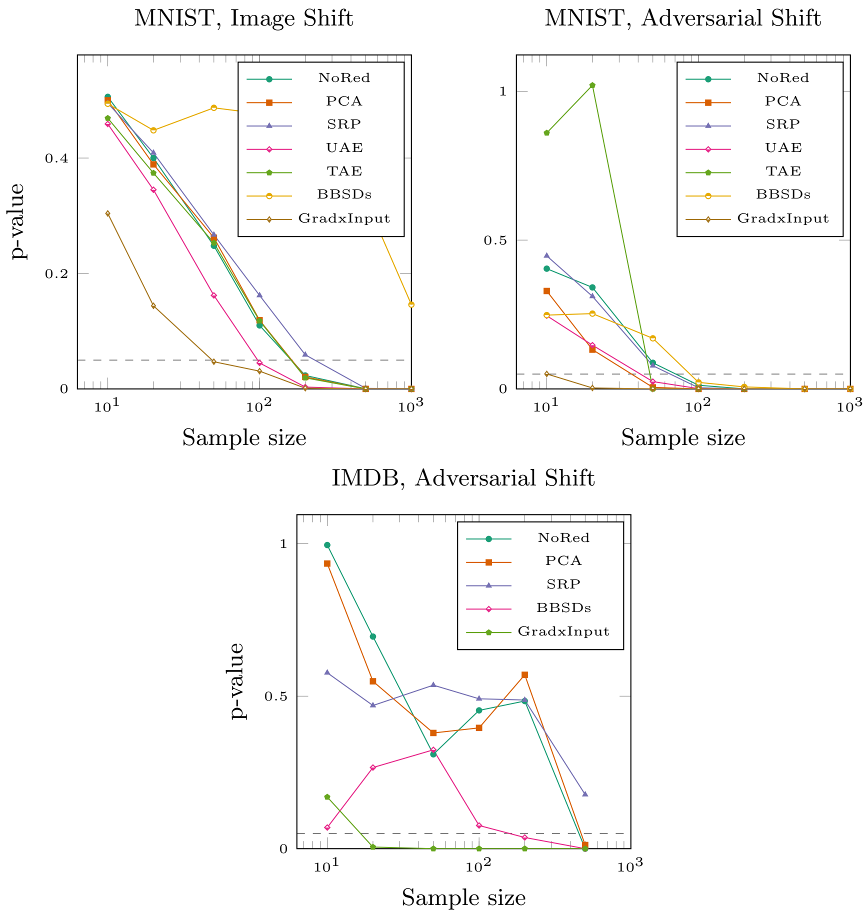

Fig. 1: **Top:** Mean p-values over 15 random runs of the MMD test for small image shift (left) and adversarial shift (right) with the MNIST dataset. In all experiments, the null hypothesis is rejected with a lower number of samples when input times gradient is used as the representation of the input. H_0 is rejected when the p-value ≤ 0.05. **Bottom:** Mean p-values for the IMDB adversarial experiment. A significant shift is detected with gradient times input at a lower sample size than any of the other representations.

4 Results

Results are shown in Figure 1. The gradient times input methods generally outperform other methods tested in sensitivity: They are able to reject H_0 with an order of magnitude fewer samples. This method therefore improves over the baseline in sensitivity of detecting drift. The gradient times input seems especially useful for detecting adversarial samples with MNIST, as it detects these with very high sensitivity compared to other methods. It also greatly outperforms other representations when used to detect adversarial shift in the text task.

5 Discussion and Future Work

While initial results show promise, additional testing is needed to fully compare results to Rabanser et al. and establish the sensitivity of the method. Our experiments differ from Rabanser et al. in a few ways: First, we only use the multivariate MMD test, as the authors found it to have similar shift detection performance to the univariate KS test which they also evaluated. Future findings should also provide results on the KS test. In addition, we perturb all test samples, rather than fractions of the test set. Thus we do not evaluate the performance when only a subset of samples has shifted. We also do not evaluate on all types of shift identified by Rabanser et al. A more definitive comparison can only be established after all tests from the original paper are assessed.

Testing on a wider variety of model architectures and datasets would also provide more opportunity to demonstrate the strong performance of this method. In further testing, models such as BERT [1] should be evaluated in the text domain for other tasks such as question answering. In addition, other types of explanations such as PatternAttribution [7] or SmoothGrad [15] can be evaluated in place of gradient times input, as they have been found to reduce noise in explanations. Comparing against other baselines [2][9] is also important to show that this method is state-of-the-art.

6 Conclusion

We demonstrate initial promising results which show improvement upon the framework employed by Rabanser et al. Representing data using explanations from the model in the form of gradient times input provides additional information about the data domain for two-sample tests, and helps improves shift detection sensitivity beyond the performance seen by the baseline.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
2. Elsahar, H., Gallé, M.: To annotate or not? predicting performance drop under domain shift. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2163–2173 (2019)
3. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
4. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research **13**(1), 723–773 (2012)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
6. Jin, D., Jin, Z., Zhou, J.T., Szolovits, P.: Is bert really robust? natural language attack on text classification and entailment. arXiv preprint arXiv:1907.11932 (2019)
7. Kindermans, P.J., Schütt, K.T., Alber, M., Müller, K.R., Erhan, D., Kim, B., Dähne, S.: Learning how to explain neural networks: Patternnet and patternattribution. arXiv preprint arXiv:1705.05598 (2017)
8. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010), `http://yann.lecun.com/exdb/mnist/`
9. Lipton, Z., Wang, Y.X., Smola, A.: Detecting and correcting for label shift with black box predictors. In: International conference on machine learning. pp. 3122–3130. PMLR (2018)
10. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA (June 2011), `http://www.aclweb.org/anthology/P11-1015`
11. Rabanser, S., Günnemann, S., Lipton, Z.C.: Failing loudly: An empirical study of methods for detecting dataset shift. arXiv preprint arXiv:1810.11953 (2018)
12. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR **abs/1910.01108** (2019), `http://arxiv.org/abs/1910.01108`
13. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 3145–3153. JMLR. org (2017)
14. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR **abs/1312.6034** (2014)
15. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)