Natural Language Processing: Detecting t€xt @ttåck$ with Robust Density Estimation

Jeanne Devineau
ENSAE
jeanne.devineau@ensae.fr

Adrien Majka
ENSAE
adrien.majka@ensae.fr

Abstract

In Natural Language Processing, detecting adversarial examples is key nowadays. A common method is to use density to detect these attacks, as adversarial examples tends to have a lower density than original ones. To upgrade the maximum likelihood estimator that is commonly used, we apply a Robust Density Estimation method which consists in using the kernel PCA and Minimum Covariance Determinant of our embeddings, inspired from [20]. We obtained relevant results with the important IMDB dataset to which we applied the transformer-based model BERT [6]. Our results with RDE showed indeed that the adversarial examples have a lower density than the original ones. Our auc is convincing enough (about 0.9) about the power of this detection model. In future research, it would be interesting to have a model that depends less upon the embeddings it is calibrated on, using the diverse variations of the BERT transformer models for example.

1 Problem Framing

1.1 Introduction

The field of Natural Language Processing (NLP) has seen significant improvements due to the application of advanced Machine Learning techniques. Tasks such as sentiment classification and text categorization have greatly benefited from these techniques, but criticisms regarding the reliance on "black-box" neural networks persist. One of the primary concerns is the sensitivity of these models to changes in input data distribution, which limits their adoption despite their high accuracy [15, 14].

To address this challenge, it is crucial to develop techniques that can detect shifts in the distribution of text and sentences [10, 5, 4]. One promising approach is to use latent representations of tokens to measure their proximity. This involves creating detectors that can identify when input samples are out of distribution or even subject to attacks, which is the focus of our paper. In addition, it is important to note that this technique can be extended to various NLP tasks beyond the detection of attacks, providing a more comprehensive solution to the problem of distributional shifts. Ultimately, the development of such techniques will help ensure the robustness and reliability of NLP models in real-world applications.

A lot of methods are being developed to recognize OOD or attacks, based on diverse metrics comparisons. For example, in [2],for an out of distribution problem, which is the same kind of problem, the method is about computing an average latent representation x and then its OOD score through the (integrated weighted rank) depth score of x with respect to the averaged in-distribution law. In another

example, in [9], they look at the input's trajectory, compared to the reference, because the distribution is different between in-sample and out-of-distribution sample.

In our article here, we will reproduce the method of [20], which has a close idea, since it compares the distribution of adversarial and normal samples, which are generally quite different.

1.2 General framework

In [20], the tool of comparison is density parametric estimation, and it's made robust with two methods : kernel PCA and Minimum Covariance Determinant. The framework is the following : given an input sample X and a label space Y, a predictive model $F : X \rightarrow Y$ and an oracle model $F^* : X \rightarrow Y$, an adversarial example x_{adv} of an input $x \in X$ satisfies : $F^*(x) = F(x) \neq F(x_{adv})$ and $C_i(x, x_{adv}) = 1$ for $i \in \{1, ..., c\}$ where C_i is an indicator function for the *i*-th constraint between the perturbed text and the original text, which is 1 when the two texts are indistinguishable with respect to the constraint. The classification model chosen is explained in section 2.

Then, they fit a parametric density estimation, to yield likelihoods of each sample, because generally the likelihood of adversaries examples is lower. Let $z \in Z \subset \mathbb{R}^D$ denote the feature. Given a generative model p_θ with mean and covariance as parameters $\theta = (\mu, \Sigma)$, we can use the features of the training samples $(Xtrain)$ to estimate the parameters. Then, novel adversarial samples, which are in the unobserved feature space, are likely to be assigned a low probability, because the model only used the normal samples for parameter estimation. For simplicity, we assume the distributions of the feature z follow a multivariate Gaussian, so we can model the class conditional probability as $p_\theta(z|y = k) \sim N(\mu_k, \Sigma_k)\ \alpha\ exp(-(z - \mu_k)^T \Sigma_k^{-1}(z - \mu k))$, where y indicates the class of a given task. Then, the maximum likelihood estimate (MLE) is given by the sample mean $\tilde{\mu}_{MLE} = \frac{1}{N} \sum_{i=1}^{n} z_i$ and sample covariance $\tilde{\Sigma}_{MLE} = \frac{1}{N} \sum_{i=1}^{n} (z_i - \tilde{\mu}_{MLE})(z_i - \tilde{\mu}_{MLE})^T$.

1.3 Robust density estimation with kPCA and MCD

However, with the previous method, accurate estimation of the parameters is difficult with finite amount of samples, especially in high dimensions (which is often the case with NLP) due to curse of dimensionality. This leads to two problems : sparse data points and fallacious features, and secondly occasional outliers that influence the parameter estimates.

Besides, using a Gaussian model distribution for the features to estimate the parameters can be misleading, because data features have more of an elliptic distribution, with thicker tails.

These two problems can be tackled using robust density estimation combining two methods : kernel PCA, proposed by [19] and Minimum Covariance Determinant (MCD) [18].

First, we use kPCA to address the issue of sparse data points and fallacious features. Intuitively, it retains the most meaningful feature dimensions, which explains the data the most, while reducing spurious features. The idea is to select top P orthogonal basis that best explain the variance of the data, thereby reducing redundant features. The mathematical formulation is the following : given N centered samples $Z_{train} \in \mathbb{R}^{N \times D} = [z_1, ...z_N]$, and a mapping function $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^{D'}$, kPCA projects the data points to the eigenvectors with the P largest eigenvalues of the covariance $\phi(Z_{train})^T \phi(Z_{train})^2$.

Besides, we need the second method,

MCD, to remedy the problem of sample-level outliers, by removing them. The method is the following : it finds a subset of $h \leq N$ samples that minimize the variance of Σ. As the determinant is proportional to the differential entropy of a Gaussian up to a logarithm (which is shown by the authors), this results in a robust covariance estimation consisting of centered data points rather than outliers.

So, to summarize, we retain relevant and informative features with the kPCA method and a robust covariance by applying MCD, on the training dataset. The choice of P and h are important to discuss. We then have robust estimated parameters, letting us evaluate the likelihood of a test sample, and categorizing the low likelihoods as adversarial examples.

1.4 Performance evaluation

To study performance of models, models detecting OOD and attacks often use the evaluation metric of the area under the receiver operating characteristic $AUROC$, and we can also look at the True Positive Rate TPR, which is the fraction of true adversarial samples out of predicted adversarial samples, or False Positive Rate FPR. The $AUROC$ measures the area under TPR vs. FPR curve. The $F1 - score$ $(f1)$ gives the harmonic mean of precision and recall. For all of these metrics, the higher they are, the better.

2 Experiments Protocol

2.1 Data

In [20], they generated adversarial examples with three datasets with data of diverse topics and length : IMDB, SST-2 and AG-News. Generating these examples require a lot of time because of all of the queries it takes (more than 40 hours). Thus, we will use directly the data of the authors, since they made it available to the public in their Github. The data has the advantage of being really clean, so we won't need to do a lot of preprocessing, simply removing brackets from the text before using the embeddings directly.

It allowed us to implement the first scenario from [20], ie we sample two disjoint subsets $S_1, S_2 \subset D$ where D is the dataset. From S_1 we keep only the successful attacks, and form a test dataset $S_1 \cup S_2$. The model will be tested on this union, while it will be trained on the rest of the dataset.

2.2 Model

For data treatment, we use the transformer-based model BERT, which is a classical NLP transformer model, fine tuned for the classification task specific to each dataset. Then, we apply the robust density estimation method, transforming our embeddings with the kernel PCA. After that, we can use the Minimum Covariance Determinant. In the following work, we use the Maximum Likelihood Estimation to compare our robust estimation to the simple MLE one. We expect different densities, the RDE showing more difference for adverse examples than the MLE one.

Then, we use our RDE method to compare adversarial and original examples from the dataset. The metric used to assess the quality of the detection is explained in the "Evaluation metrics" section.

2.3 Attacks

Finally, to test how our RDE model responds to attacks, we use different types of attacks : there are Textfooler [12] attacks, Probability Weighted Word Saliency (PWWS) [17] attacks and BAE attacks [8].

2.4 Evaluation metrics

The primary metrics we looked was the AUC, as it is a metric quite robust to class imbalance, which could occur during our experience as we kept only successful attacks from

the model. It gives a quite visual understanding of how the model is working. Moreover, and contrary to the F1 score and true positive rate (TPR), it encompass global information on false positive rate *and* true positive rate (TPR), while F1-score and TPR needs FPR to be fixed. This being said, for the sake of completeness we also calculated F1-score and TPR for the different experiments we conducted, with a FPR to 0.1 similarly to [20].

2.5 Implementation details

We directly implemented the algorithm of attack detection on the dataset proposed by [20], as we encountered some difficulties in implementation when trying to run the attacks (Textfooler, BAE and PWWS). Most of the algorithmic work, notably the implementation of MCD, was done thanks to Scikit-learn library [13]. Most of the rest of the work was straight calculation feasible with the classic numpy library. What asked for much more work was to familiarize ourselves with deep learning library. Hopefully, we quickly discovered the SentenceTransformers library [16], with which we made all the embeddings.

3 Results

Note that the code is available at `https://github.com/Adrien-Mcode/Text-adversarial-attack-NLP3A.git`

3.1 Qualitative results

First of all, let's discuss the results of the Robust Density Estimation method, compared to a simple MLE method, both tested on the IMDB dataset.

The graph 1 shows the result of the density estimations when we use the Maximum Likelihood estimator, without the robust density estimation. As we see, there is not much difference in the density. It is also very large which give us a hint on the scale of the first eigenvalue of the covariance matrix and the conditionning of the matrix which is empirically pretty bad.

Then, we apply the kernel PCA and Minimum Covariance Determinant to have a more robust estimation. As we see in the following graph 2, there is now a difference in the density, and the density is much more concentrated around big cluster of point. We can see this very well with the graph 3

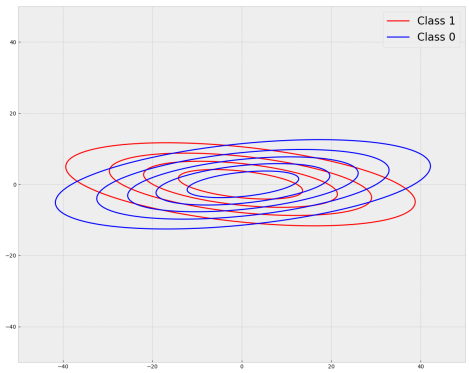

FIGURE 1 – Normal multivariate distribution estimated via MLE

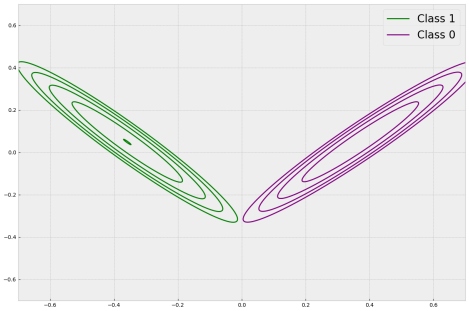

FIGURE 2 – Normal multivariate distribution estimated via RDE

Indeed, we see that the density with the MLE estimation is very close for almost every point in the dataset, while for the RDE estimation the density mass is concentrated around cluster with a lot of point and rapidly decrease outside of it. This is of course an advantage to discriminate out of sample attack.

Knowing this, we can use the RDE method to differentiate the adversarial examples from the original ones of our IMDB dataset, by looking at the density, expecting the density of adversarial examples to be lower. In the following graph, we plot the density of points from the adversarial examples, and the one of original text samples. We see a difference in the graphs indeed.

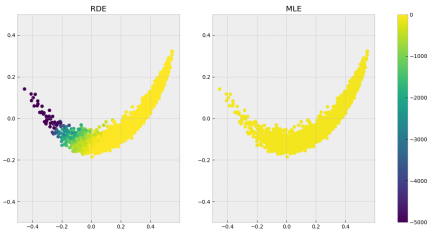

FIGURE 3 – Comparison of density at each point with respect to MLE and RDE estimation

3.2 Model performance

Let's now look at our model performance display in 4 in terms of metric. We can see here that the RDE method is clearly above in terms of performance than the MLE, for all metrics considered. Moreover, even when the MLE performs poorly, the RDE is still efficient. It's worth noticing also the stability of metrics across the different attacks, notably the true positive rate, which might let us think that RDE is well suited as a defense mechanism against all attacks studied here.

4 Discussion/Conclusion

In conclusion, the method used in this study has shown promising and stable results in terms of robustness against attacks, while remaining relatively simple and easy to implement. However, there are still opportunities for further improvement. One area for potential enhancement is testing the method in a multimodal setting [7], where the input data includes information from multiple modalities such as text, images, and audio. Additionally, testing the method in the context of sequence generation tasks [3, 11, 1] would be valuable, as this presents a unique set of challenges that may require modifications to the current approach.

Another aspect that could be improved is the dependence of the BERT model on the embeddings used. As such, future research could focus on exploring alternative embedding techniques that may provide better performance in terms of robustness against attacks. Overall, these potential enhancements can help to further strengthen the reliability and efficacy of the current method, opening up new possibilities for its application in various domains.

[1]

Références

[1] Pierre COLOMBO, Chloe CLAVEL et Pablo PIANTANIDA. « A Novel Estimator of Mutual Information for Learning to Disentangle Textual Representations ». In : *() ACL 2021* (2021).

[2] Pierre COLOMBO et al. « Beyond Mahalanobis-Based Scores for Textual OOD Detection ». In : (2022). DOI : 10.48550/ARXIV.2211.13527. URL : https://arxiv.org/abs/2211.13527.

[3] Pierre COLOMBO* et al. « Affect-driven dialog generation ». In : *NAACL 2019* (2019).

[4] Maxime DARRIN, Pablo PIANTANIDA et Pierre COLOMBO. « Rainproof: An Umbrella To Shield Text Generators From Out-Of-Distribution Data ». In : *arXiv preprint arXiv:2212.09171* (2023).

[5] Maxime DARRIN et al. « Unsupervised Layer-wise Score Aggregation for Textual OOD Detection ». In : (2023). Publisher: arXiv Version Number: 1. DOI : 10.48550/ARXIV.2302.09852. URL : https://arxiv.org/abs/2302.09852.

[6] Jacob DEVLIN et al. « BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding ». en. In : *Proceedings of the 2019 Conference of the North*. Minneapolis, Minnesota : Association for Computational Linguistics, 2019, p. 4171-4186. DOI : 10.18653/v1/N19-1423. URL : http://aclweb.org/anthology/N19-1423.

[7] Alexandre GARCIA* et al. « From the token to the review: A hierarchical multimodal approach to opinion mining ». In : *EMNLP 2019* (2019).

[8] Siddhant GARG* et Goutham RAMAKRISHNAN*. « BAE: BERT-based Adversarial Examples for Text Classification ». In : *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online : Association for Computational Linguistics, nov. 2020, p. 6174-6181. DOI : 10.18653/v1/2020.emnlp-main.498. URL : https://aclanthology.org/2020.emnlp-main.498.

[9] Eduardo Dadalto Câmara GOMES et al. *A Functional Perspective on Multi-Layer Out-of-Distribution Detection*. 2023. URL : https://openreview.net/forum?id=gyTuMfkOney.

1. * means that both author contributes equally

	bae			pruthi			pwws			textfooler		
	auc	f1	tpr	auc	f1	tpr	auc	f1	tpr	auc	f1	tpr
MLE	0.730	0.220	0.325	0.691	0.920	0.267	0.610	0.221	0.190	0.621	0.195	0.207
RDE	0.909	0.770	0.810	0.891	0.920	0.777	0.868	0.685	0.764	0.900	0.686	0.805

	tf-adj		
‘	auc	f1	Tpr
MLE	0.746	0.217	0.350
RDE	0.907	0.927	0.806

FIGURE 4 – Performance for the RDE and MLE model on different metrics and different kinds of attacks

[10] Nuno M GUERREIRO et al. « Optimal Transport for Unsupervised Hallucination Detection in Neural Machine Translation ». In : *arXiv preprint arXiv:2212.09631* (2023).

[11] Hamid JALALZAI* et al. « Heavy-tailed representations, text polarity classification & data augmentation ». In : *NeurIPS 2020* (2020).

[12] Di JIN et al. *Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment*. arXiv:1907.11932 [cs]. Avr. 2020. URL : http://arxiv.org/abs/1907.11932.

[13] Fabian PEDREGOSA et al. « Scikit-learn: Machine Learning in Python ». In : *Journal of Machine Learning Research* 12.85 (2011), p. 2825-2830. URL : http://jmlr.org/papers/v12/pedregosa11a.html.

[14] Marine PICOT et al. « A Simple Unsupervised Data Depth-based Method to Detect Adversarial Images ». In : (2023).

[15] Marine PICOT et al. « Adversarial Attack Detection Under Realistic Constraints ». In : (2023).

[16] Nils REIMERS et Iryna GUREVYCH. « Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks ». In : (2019). Publisher: arXiv Version Number: 1. DOI : 10.48550/ARXIV.1908.10084. URL : https://arxiv.org/abs/1908.10084.

[17] Shuhuai REN et al. « Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency ». In : *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy : Association for Computational Linguistics, juill. 2019, p. 1085-1097. DOI : 10.18653/v1/P19-1103. URL : https://aclanthology.org/P19-1103.

[18] Peter J. ROUSSEEUW. « Least Median of Squares Regression ». en. In : *Journal of the American Statistical Association* 79.388 (déc. 1984), p. 871-880. ISSN : 0162-1459, 1537-274X. DOI : 10.1080/01621459.1984.10477105. URL : http://www.tandfonline.com/doi/abs/10.1080/01621459.1984.10477105.

[19] Bernhard SCHÖLKOPF, Alexander SMOLA et Klaus-Robert MÜLLER. « Kernel principal component analysis ». In : *Artificial Neural Networks — ICANN'97*. Sous la dir. de Gerhard GOOS et al. T. 1327. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 1997, p. 583-588. ISBN : 978-3-540-63631-1 978-3-540-69620-9. DOI : 10.1007/BFb0020217. URL : http://link.springer.com/10.1007/BFb0020217.

[20] KiYoon YOO et al. « Detection of Adversarial Examples in Text Classification: Benchmark and Baseline via Robust Density Estimation ». In : *Findings of the Association for Computational Linguistics: ACL 2022*. Dublin, Ireland : Association for Computational Linguistics, mai 2022, p. 3656-3672. DOI : 10.18653/v1/2022.findings-acl.289. URL : https://aclanthology.org/2022.findings-acl.289.