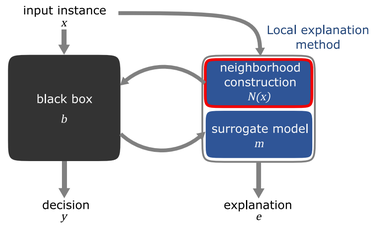
XPROAX – local eXplanations for text classification with PROgressive neighborhood ApproXimation

Yi Cai[1], Arthur Zimek[2], Eirini Ntoutsi[1,3]

DSAA 2021 IEEE | Cosponsors ACM/ ASA/ CCF

Local explanation methods



Research challenges

- Existing researches (e.g. LIME) using word dropping will lead to incomplete sentences, which may not be the optimal neighborhood.
- Other methods (e.g. XSPELLS) sampling randomly in a latent space can result in low-quality neighborhoods.

XPROAX: overview

Idea: Deploy a generative autoencoder to construct a better neighborhood (semantically meaningful and grammatically correct); use landmarks from a corpus to approximate the neighborhood and follow the data manifold.

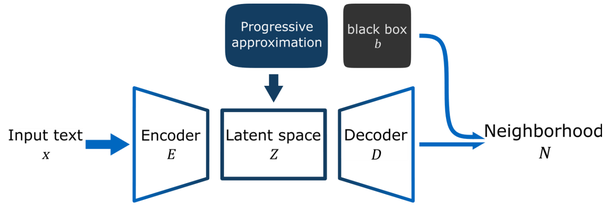
a) Map the input text x into the latent space Z via the encoder E

b) Generate neighboring texts in the latent space Z using **progressive neighborhood approximation**

c) Reconstruct neighboring texts from latent vectors with the decoder D

d) Label the neighboring texts with the black box

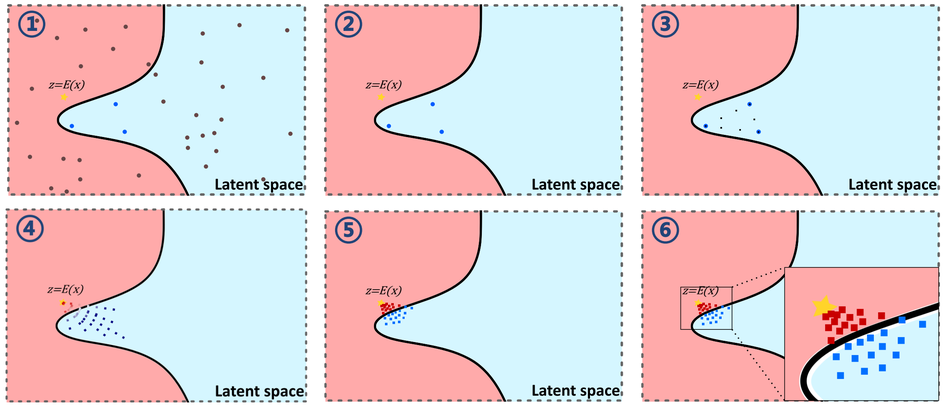
e) Train a surrogate model with the neighboring texts $N(x)$ for explanations



XPROAX: Explanation

XPROAX provides explanations, which consist of four components:

1. *Intrinsic* words – words in the input x;
2. *Extrinsic* words – words only appeared in the neighborhood N;
3. Top-k factuals
4. Top-k counterfactuals



XPROAX: Progressive neighborhood approximation

① Map the input x into the latent space $z = E(x)$

② Select the k-closest counterfactuals from a corpus as **landmarks** based on the latent distances

③ 1st interpolation: interpolate between the landmarks to better fill the gap between them to cover the decision boundary

④ 2nd interpolation: interpolate between the target point z and the counterfactuals generated during the first interpolation

⑤ Repeat step 3 and step 4 until no closer counterfactual to the input text can be found

⑥ Select the nearest points and reconstruct the texts from the latent vectors as the neighborhood of x

Experimental results – Qualitative evaluation

- Qualitative evaluation illustrates the usefulness of explanations intuitively.
- In the given example below, the term "not" as an extrinsic word contributing to the opposite sentiment proves that the model is able to handle a negation context, and the misclassification is mainly caused by the term "n't".

Input 2	fries are n't worth coming back .	Random Forest $b(\cdot)$: **positive**, Dataset: **Yelp**
XPROAX	**Saliency**: fries are n't worth coming back .	**Extrinsic words**[a]: not perfect
	Factuals: 1) the fries were n't worth coming. 2) _unk_[b] are n't worth going back. 3) the fries were worth coming back. 4) the fries were worth going back. 5) you do n't be worth coming.	**Counterfactuals**: 1) _unk_ do n't bother in back. 2) _unk_ do n't bother going back. 3) _unk_ do n't be anybody back. 4) a few fries were definately coming back. 5) _unk_ do n't be anybody.
XSPELLS	**Factuals**: 1) it seems well they did 2) and i feel like on service 3) dave is excellent 4) everything we will get 5) all i hung up is nice **Common words in factuals**: seems (0.091), well (0.091), feel (0.091)	**Counterfactuals**: 1) both to die 2) all else s 3) every i may 4) who makes me money last 5) all were nt pricey **Common words in counterfactuals**: die (0.111), else (0.111), every: (0.111)
LIME	**Saliency**: fries are n't worth coming back .	

[a]Words with high importance that only appeared in the neighbors (not appeared in the input text).
[b]Generic unknown word token for words out of the vocabulary.

Experimental results – Quantitative evaluation

Dataset and Model	Explaining Method	Confidence Drop	Avg. Confidence Drop per op	$\Delta\eta$ (0.3-0.1)
Yelp & RF	baseline	0.247 ± 0.31	0.179 ± 0.24	/
	LIME	0.364 ± 0.29	0.297 ± 0.26	**+0.213**
	XSPELLS	0.132 ± 0.26	0.170 ± 0.27	+0.032
	XPROAX	**0.740 ± 0.22**	**0.417 ± 0.33**	+0.153
Yelp & DNN	baseline	0.136 ± 0.32	0.094 ± 0.23	/
	LIME	0.564 ± 0.46	**0.348 ± 0.44**	**+0.230**
	XSPELLS	0.084 ± 0.26	0.102 ± 0.27	-0.014
	XPROAX	**0.825 ± 0.35**	0.302 ± 0.43	+0.206
Amazon & RF	baseline	0.163 ± 0.19	0.118 ± 0.14	/
	LIME	0.209 ± 0.18	0.201 ± 0.16	**+0.185**
	XSPELLS	0.048 ± 0.13	0.058 ± 0.14	+0.037
	XPROAX	**0.506 ± 0.20**	**0.354 ± 0.21**	+0.126
Amazon & DNN	baseline	0.287 ± 0.27	0.209 ± 0.21	/
	LIME	0.424 ± 0.27	0.238 ± 0.17	+0.156
	XSPELLS	0.095 ± 0.18	0.122 ± 0.18	+0.037
	XPROAX	**0.665 ± 0.21**	**0.298 ± 0.25**	**+0.164**

- Editions of inputs following explanations provided by XPROAX have the largest effect on the prediction in all experimental settings.
- XPROAX outperforms the two competitors in terms of compactness (confidence drop per operation) in 3 settings out of 4.
- The comparison to XSPELLS shows that the sampling strategy in the latent space will have a significant impact on the quality of final explanations.

Conclusion

- ✚ The experiments, both qualitatively and quantitatively, show that XPROAX outperforms state-of-the-art methods.
- ✚ The quality of neighborhoods affects final explanations.
- ✚ Explanations on text classifiers do not need to be limited by the words that appeared in the input; extrinsic words can also contribute to the understanding.
- ✚ In comparison to XSPELLS, the careful construction of the neighborhood overcomes the weakness of randomly sampling in the latent space.

Acknowledgements

Niedersächsisches Ministerium für Wissenschaft und Kultur

The first author is supported by the State Ministry of Science and Culture of Lower Saxony, within the PhD program ``Responsible Artificial Intelligence in the Digital Society''.

[1]L3S Research Center, Hannover, Germany, [2]Department of Mathematics and Computer Science (IMADA), University of Southern Denmark (SDU), Odense, Denmark, [3]Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany

Contact: **cai@l3s.de**

Code available at: https://github.com/caiy0220/XPROAX