NC STATE UNIVERSITY

GHOST

Yedida, R., & Menzies, T. (2021). On the value of oversampling for deep learning in software defect prediction. *IEEE Transactions on Software Engineering*.

Rahul Yedida, NC State

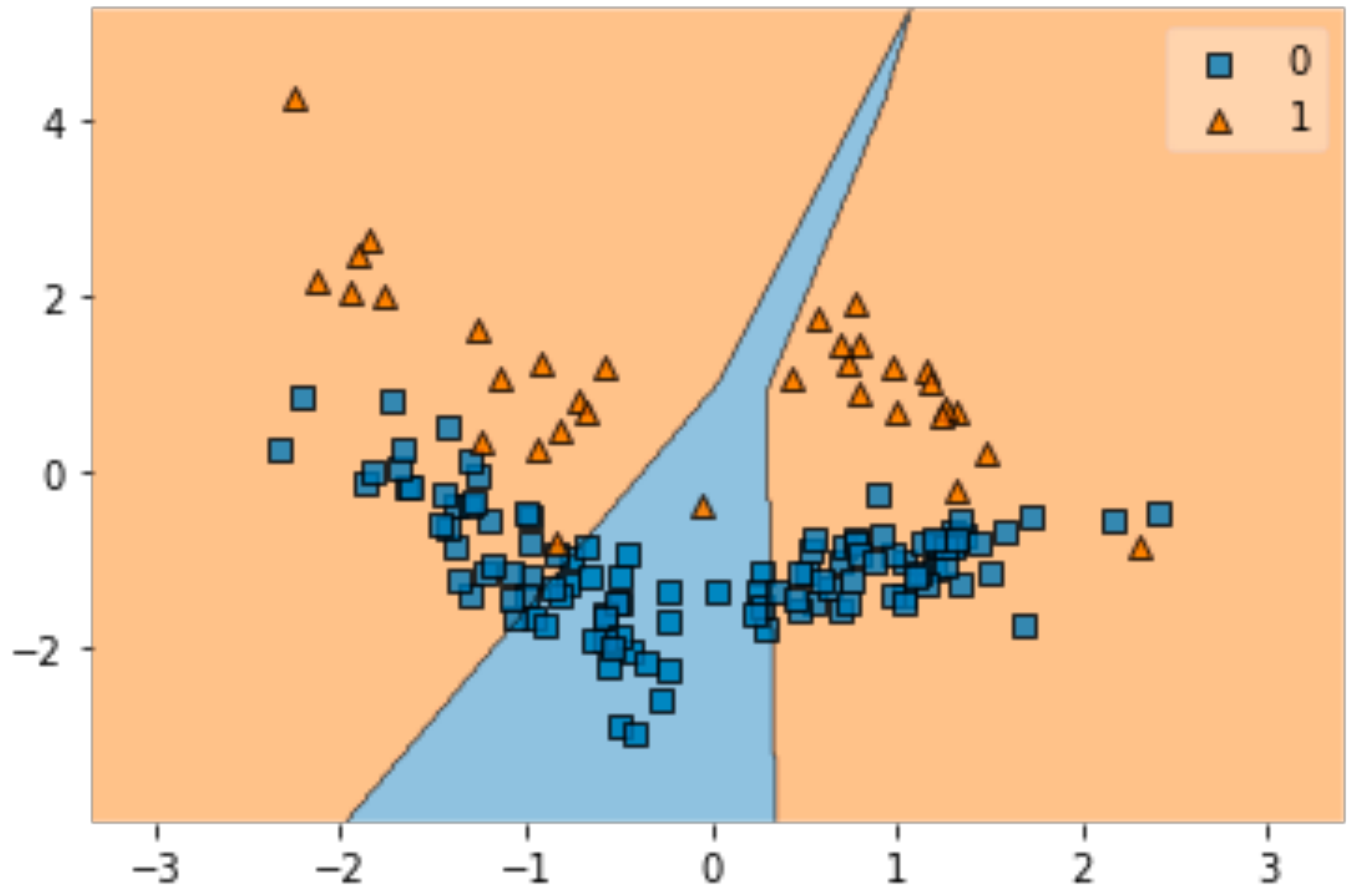
Tim Menzies, NC State

What is the problem?

Class imbalance

- Defect prediction: given static code features about files, can we predict if they are buggy?
- Defect prediction datasets have a high degree of class imbalance.

Project	Train versions	Test versions	Training Buggy %	Test Buggy %
ivy	1.1, 1.4	2.0	22	11
lucene	2.0, 2.2	2.4	53	60
poi	1.5, 2.0, 2.5	3.0	46	65
synapse	1.0, 1.1	1.2	20	34
velocity	1.4, 1.5	1.6	71	34
camel	1.0, 1.2, 1.4	1.6	21	19
jEdit	3.2, 4,0, 4.1, 4.2	4.3	23	2
log4j	1.0, 1.1	1.2	29	92
xalan	2.4, 2.5, 2.6	2.7	38	99
xerces	1.2, 1.3	1.4	16	74

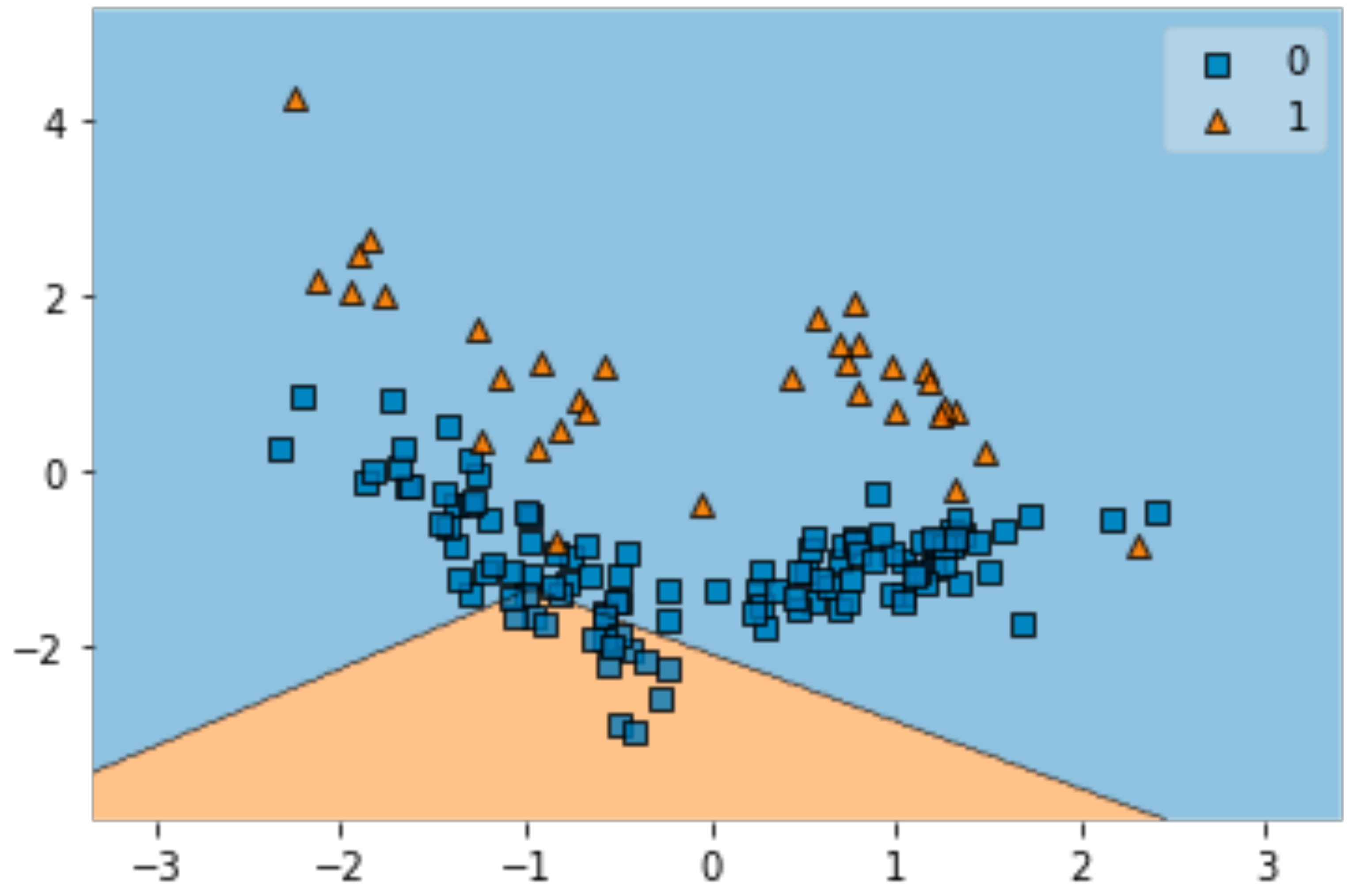


Solution 0.5

"Oversampling"

- If n is the fraction of samples with class c0:

$$\hat{\mathscr{L}}(y_i, \hat{y}_i) = \frac{w_i}{n} \sum_{\substack{i=1 \\ y_i = c_0}}^{m} \mathscr{L}(y_i, \hat{y}_i) + \sum_{\substack{i=1 \\ y_i \neq c_0}}^{m} \mathscr{L}(y_i, \hat{y}_i)$$



Solution 0.5

"Oversampling"

	deep learning a	oversampling b	SMOTE c	fuzzy sampler d	tuning e	**Median** (50th)	**IQR**	●= median and lines shows IQR
#1	✓					0	11	●
#2	✓	✓				0	4	●

Solution 0.6

Better oversampling

- Use SMOTE + hyper-parameter optimization (Fu & Menzies, 2017; Menzies et al., 2018) using DODGE (Agrawal et al., 2019)
- See also: using conditional WGANs to resample data (Shu et al., 2022)

Fu, Wei, and Tim Menzies. "Easy over hard: A case study on deep learning." *Proceedings of the 2017 11th joint meeting on foundations of software engineering*. 2017.
Menzies, Tim, et al. "500+ times faster than deep learning:(a case study exploring faster methods for text mining stackoverflow)." *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*. IEEE, 2018.
Agrawal, Amritanshu, et al. "How to “dodge” complex software analytics." *IEEE Transactions on Software Engineering* 47.10 (2019): 2182-2194.
Shu, Rui, et al. "Dazzle: Using Optimized Generative Adversarial Networks to Address Security Data Class Imbalance Issue." *arXiv preprint arXiv:2203.11410* (2022).

Solution 0.6

Better oversampling

	a deep learning	b oversampling	c SMOTE	d fuzzy sampler	e tuning	**Median** (50th)	**IQR**	●= median and lines shows IQR
#1	✓					0	11	●
#2	✓	✓				0	4	●
#3	✓	✓	✓	✓		0	15	●-
#4	✓	✓	✓		✓	0	29	●——

Why does it not work?

Boundary engineering

- Insight #1: the decision boundary is too close to the data samples.
- How to push it away?
    - Add in samples around each point: fuzzy sampling

bit.ly/fuzzy-sampling

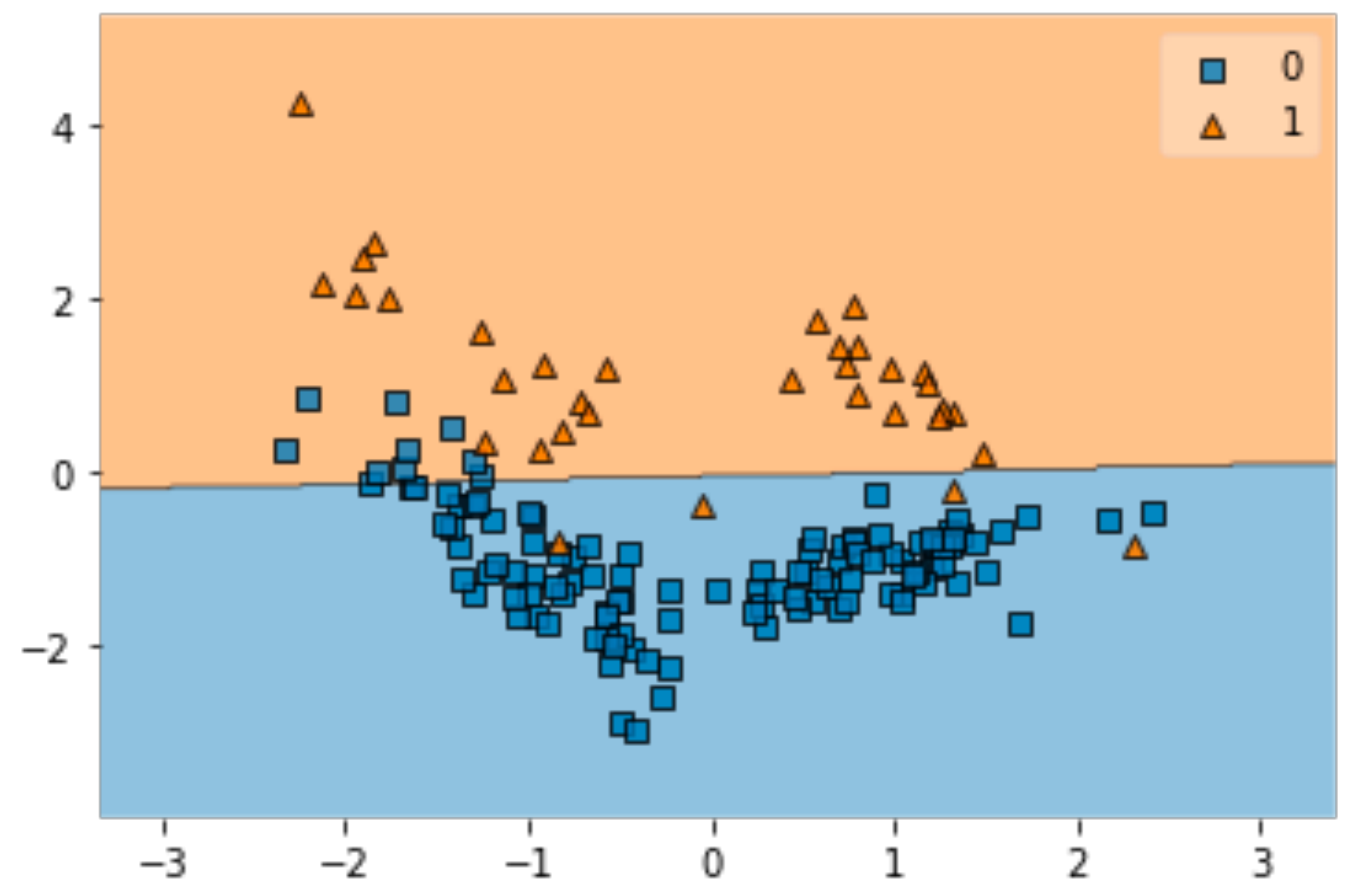
```
def fuzz_data(X, y, radii=(0., 1.5, .5)):
    idx = np.where(y == 1)[0]
    frac = len(idx) * 1. / len(y)
    print('debug: weight =', 1./frac)

    fuzzed_x = []
    fuzzed_y = []

    for row in X[idx]:
        for i, r in enumerate(np.arange(*radii)):
            for j in range(int((1./frac) / pow(2., i))):
                fuzzed_x.append([val - r for val in row])
                fuzzed_x.append([val + r for val in row])
                fuzzed_y.append(1)
                fuzzed_y.append(1)

    return np.concatenate((X, np.array(fuzzed_x)), axis=0), np.concatenate((y, np.array(fuzzed_y)))
```

Solution 0.8

Add in fuzzy sampling

- We're doing better!

	deep learning a	oversampling b	SMOTE c	fuzzy sampler d	tuning e	**Median** (50th)	**IQR**	●= median and lines shows IQR
#1	✓					0	11	
#2	✓	✓				0	4	
#3	✓	✓	✓	✓		0	15	
#4	✓	✓	✓		✓	0	29	
#5	✓		✓	✓	✓	51	25	
#6	✓	✓		✓	✓	51	25	
#7	✓	✓	✓	✓	✓	51	25	

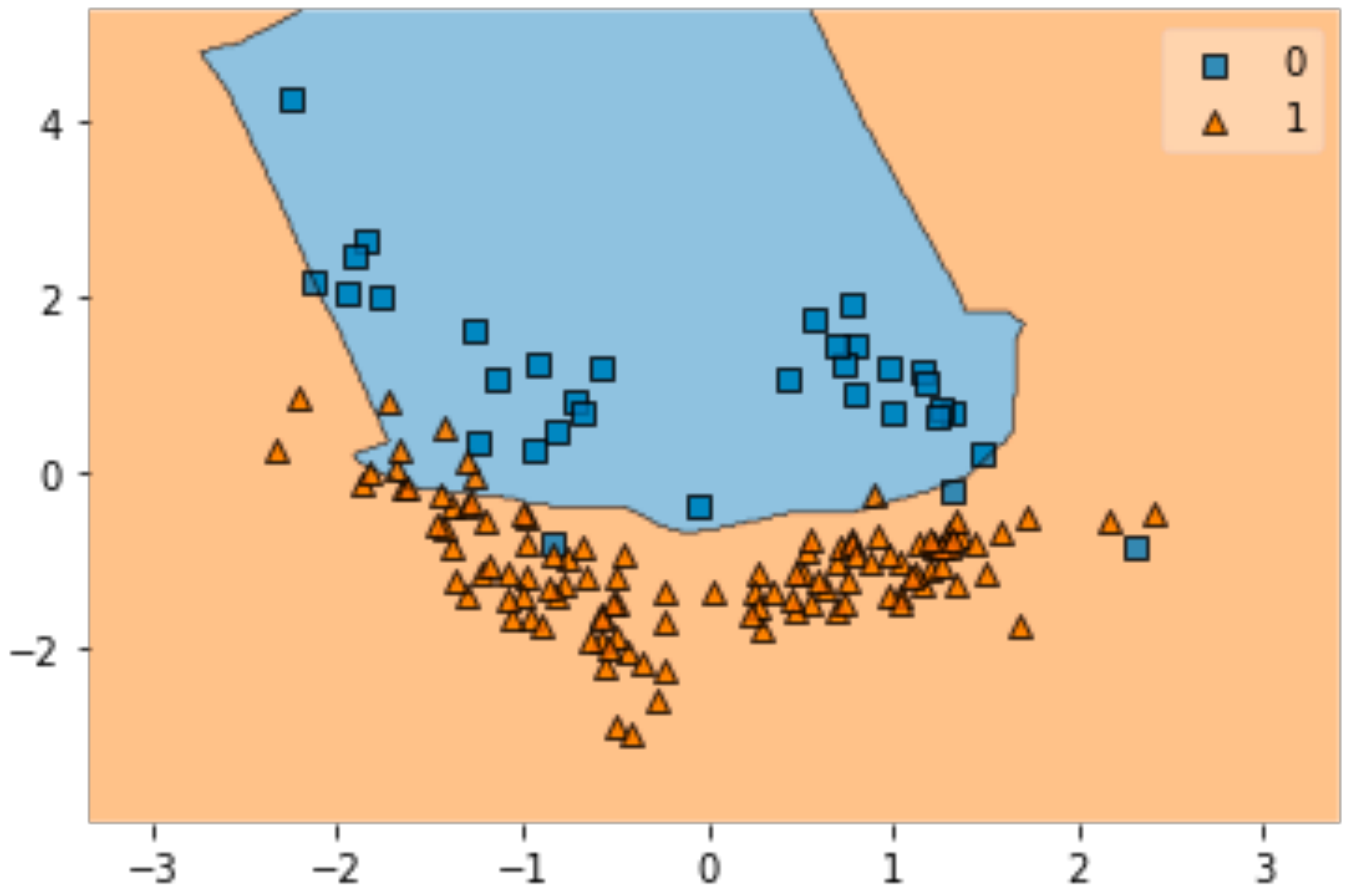


Solution 1.0

GHOST

- Insight #2: we can do the same for the majority samples!

	deep learning a	oversampling b	SMOTE c	fuzzy sampler d	tuning e	Median (50th)	IQR	•= median and lines shows IQR
#1	✓					0	11	
#2	✓	✓				0	4	
#3	✓	✓	✓	✓		0	15	
#4	✓	✓	✓		✓	0	29	
#5	✓		✓	✓	✓	51	25	
#6	✓	✓		✓	✓	51	25	
#7	✓	✓	✓	✓	✓	51	25	
GHOST = #8	✓	✓	✓	✓✓	✓	80	25	

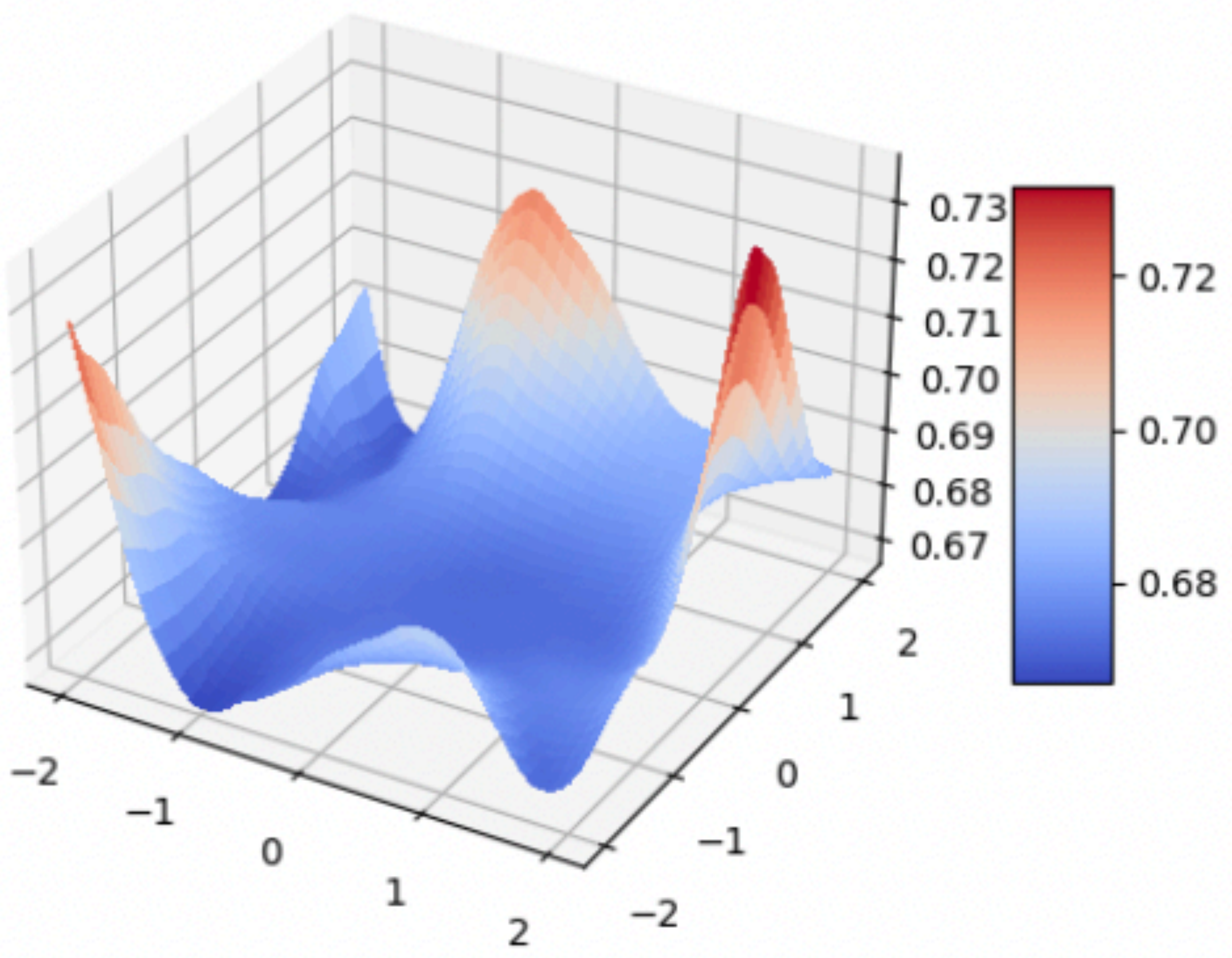


Summary

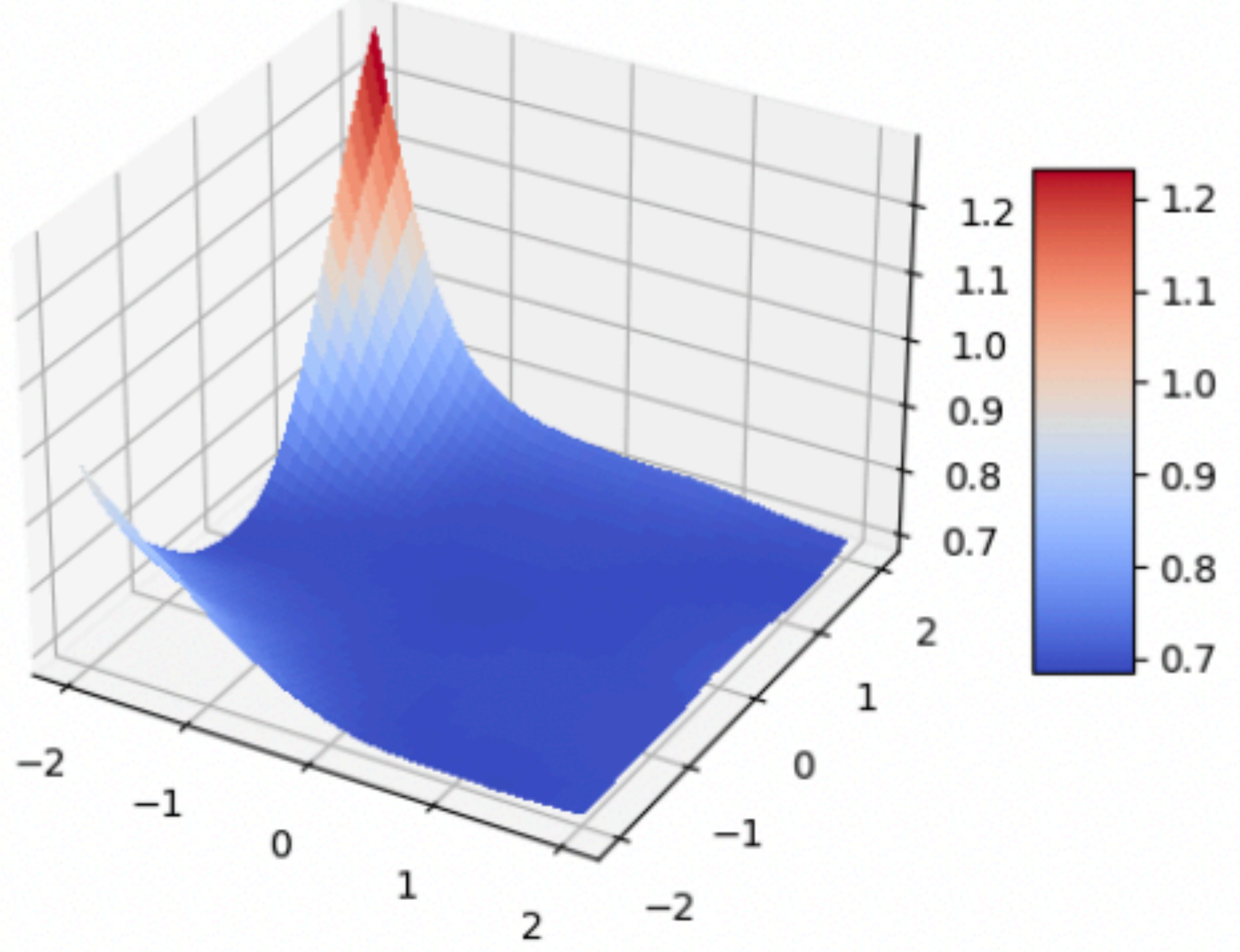
- We oversample (fuzzy sampling)
- We oversample again (fuzzy sampling #2)
- We oversample yet again (SMOTE)
- We also use weighted loss functions

By the way:

Fuzzy sampling improves beta-smoothness



(a) Before fuzzy sampling



(b) After fuzzy sampling

Thank you!

GHOST paper	http://tiny.cc/ghost-paper
Weighted fuzzy oversampling	bit.ly/fuzzy-sampling
These slides	bit.ly/ghost-slides
Contact me	ryedida@ncsu.edu