Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

data science

Density-based weighting for imbalanced regression

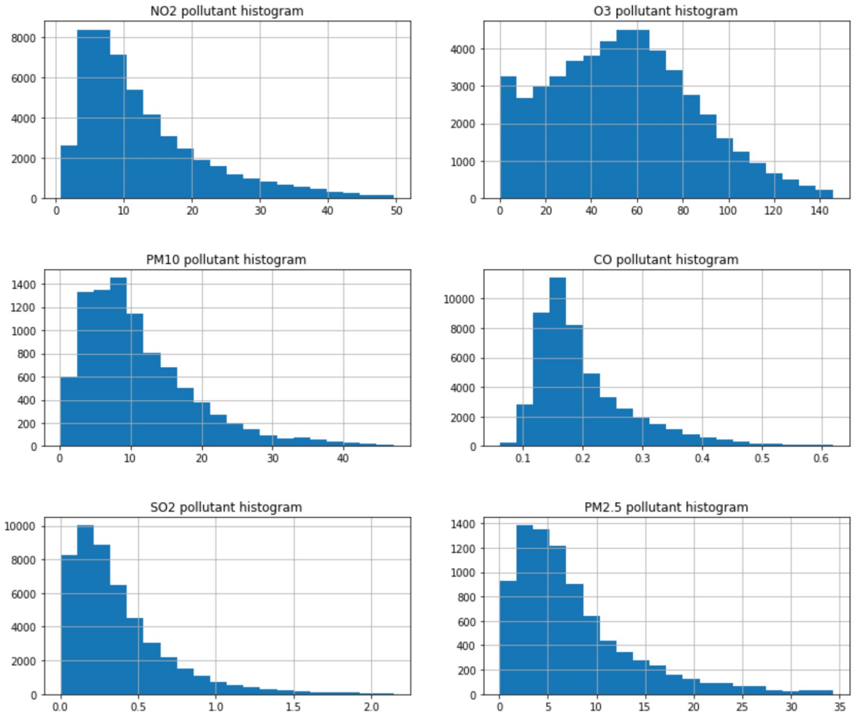
Michael Steininger

Outline

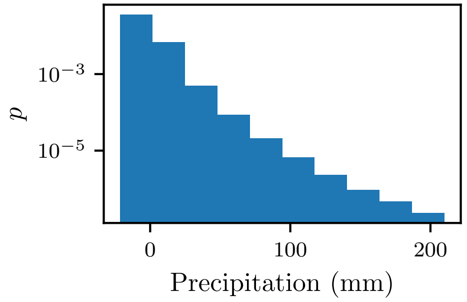
- Motivation
- Learning with Imbalanced Data
- Resampling Approaches
  - SMOTE for Regression [Torgo et al. 2013]
  - SMOGN [Branco et al. 2017]
  - Geometric SMOTE [Douzas et al. 2019; Camacho et al. 2022]
- Metric: SERA [Ribeiro & Moniz 2020]
- Cost-sensitive Learning
  - DenseWeight/-Loss [Steininger et al. 2021]
  - Label Distribution Smoothing & Feature Distribution Smoothing [Yang et al. 2021]
  - Balanced MSE [Ren et al. 2022]
- Conclusion

Motivation

- Machine learning algorithms typically expect uniform target distributions
- Models trained on imbalanced data are biased towards “common” cases
  - Rare cases often most interesting (e.g. extreme precipitation)
- How can we improve performance for rare cases when training on imbalanced data?

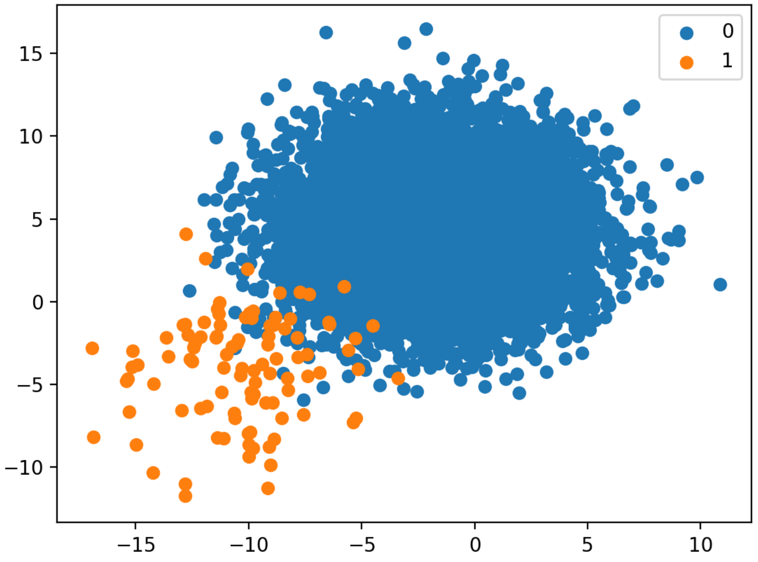


[https://ada-pandas.github.io/index.html]



Learning with Imbalanced Data

- Well researched topic for classification tasks
- Numerous approaches exist based on:
  - Resampling: SMOTE [Chawla et al. 2002], ADASYN [He et al. 2008], …
  - Cost-sensitive learning: inverse class frequency, …



[https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/]

- Approaches for regression tasks:
  - Resampling: SmoteR [Torgo et al. 2013], SMOGN [Branco et al. 2017], Geometric SMOTE for regression [Camacho et al. 2022]
  - Metric: SERA [Ribeiro & Moniz 2020]
  - Cost-sensitive learning: DenseLoss [Steininger et al. 2021], Label/Feature Distribution Smoothing [Yang et al. 2021], Balanced MSE [Ren et al. 2022]

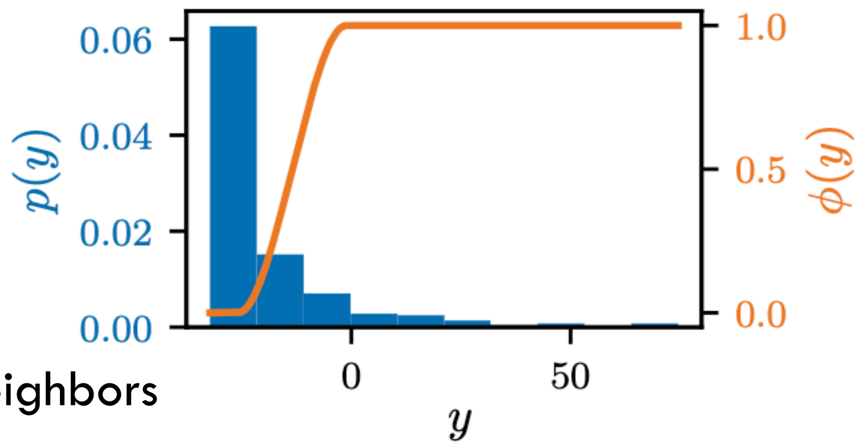
Resampling Approaches

- Basic Idea: Alter target distribution by resampling the dataset
- Basic Techniques:
  - Undersample majority class(es)
  - Oversample minority class(es) i.e. generate additional (synthetic) samples from existing samples
- Pros:
  - Independent of machine learning model
- Cons:
  - Undersampling may remove helpful information
  - Oversampling may amplify overfitting and may add noise

SMOTE for Regression [Torgo et al. 2013]

- As the name suggests: Adaptation of SMOTE for regression tasks

- Approach
  - Obtain a relevance function $\phi(Y): Y \rightarrow [0, 1]$
  - Bin samples into rare and normal based on ϕ
  - Generate new synthetic samples in rare bin(s)
    - Interpolation between a sample and one of its nearest neighbors
  - Remove samples from normal bin



→ More balanced dataset

SMOGN
[Branco et al. 2017]

- Builds upon SMOTE for Regression

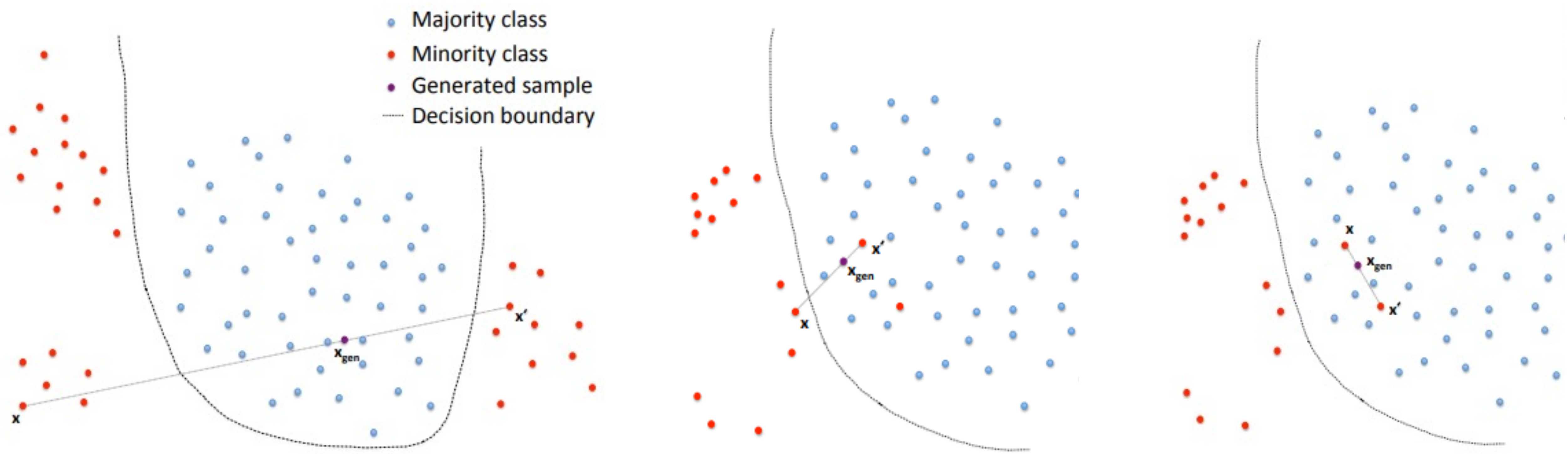
- Adds a Gaussian Noise strategy in addition to the interpolation strategy for Oversampling
  - If nearest neighbors too far away from seed sample:
    - Generate a new sample by adding Gaussian Noise to seed sample

- Tends to perform better than SMOTE for regression

Geometric SMOTE

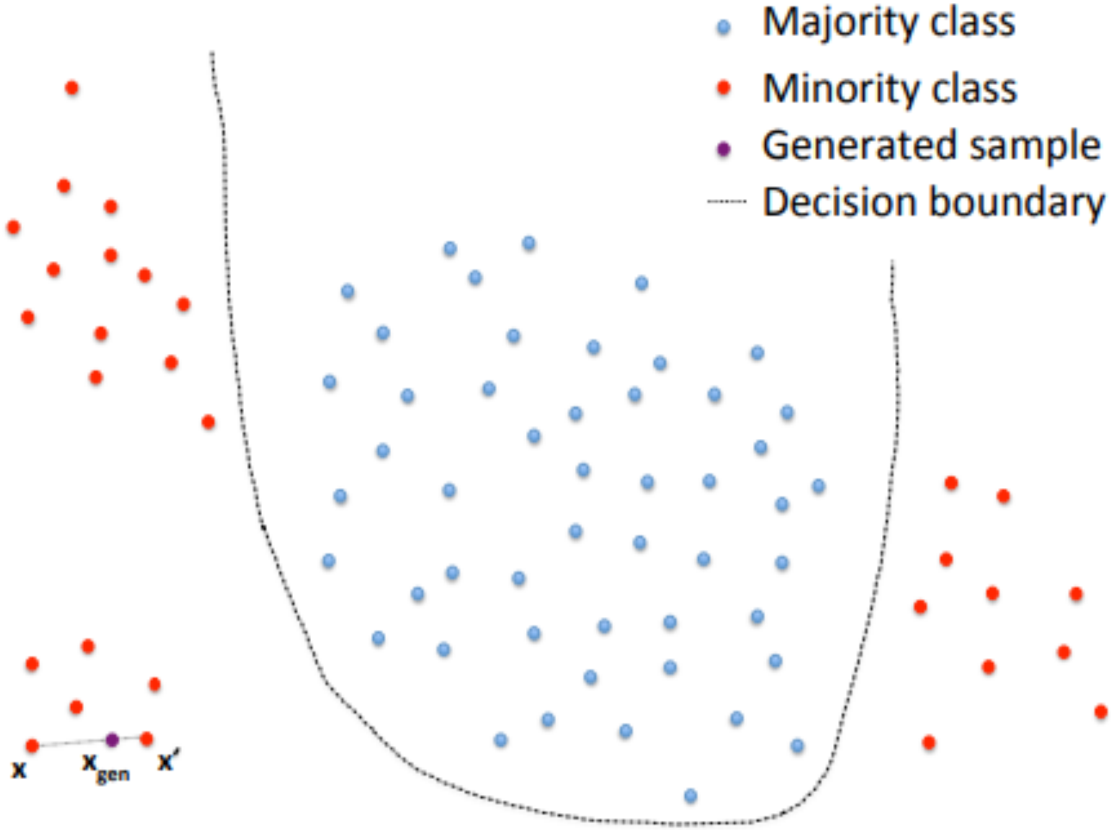
[Douzas et al. 2019; Camacho et al. 2022]

- Wants to address issues with SMOTE:
  - Generation of noisy samples



Geometric SMOTE
[Douzas et al. 2019; Camacho et al. 2022]

- Wants to address issues with SMOTE:
  - Generation of too similar samples



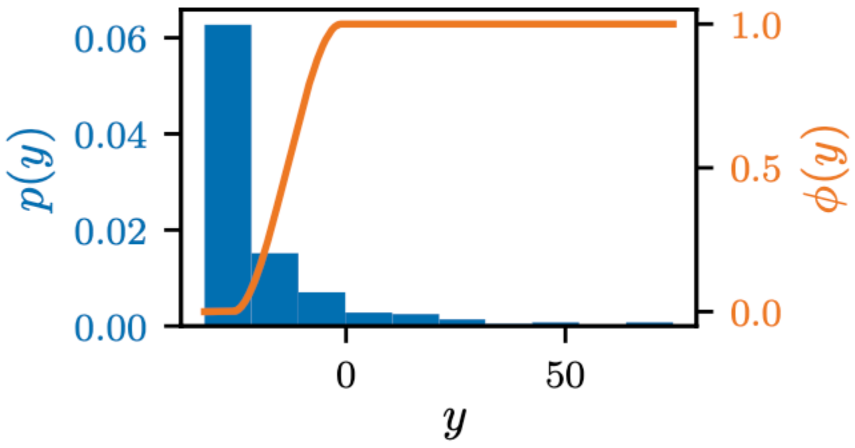
Geometric SMOTE
[Douzas et al. 2019; Camacho et al. 2022]

- Basic Idea:
  - Define a **safe area** as a **geometric region** around each minority sample
    - Samples generated in this safe area are **not noisy**
  - Expand safe area to increase variety of generated samples

- Camacho et al. 2022 adapted this approach for regression tasks

SERA
[Ribeiro & Moniz 2020]

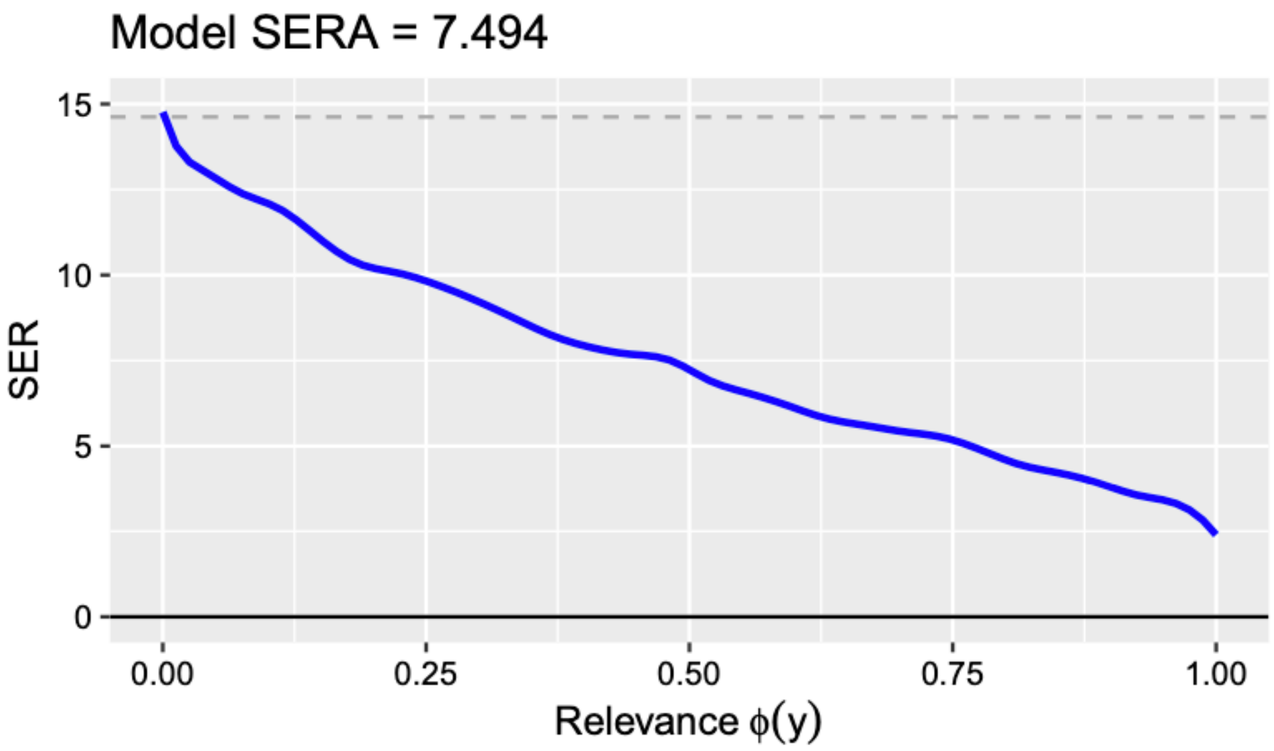
- Squared error-relevance area (SERA)
- assesses the effectiveness of models for the prediction of extreme values while penalising severe model bias



- $SER_t = \sum_{i \in D^t} (\hat{y}_i - y_i)^2$
- $SERA = \int_0^1 SER_t \, dt = \int_0^1 \sum_{i \in D^t} (\hat{y}_i - y_i)^2 \; dt$

SERA
[Ribeiro & Moniz 2020]

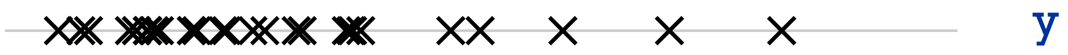
Fig. 8 An example of the squared error-relevance area (*SERA*) metric for an artificial model, based on the integration of Squared Error-Relevance (SER_t) for cutoff relevance $\phi(.)$ values t. The grey dashed line depicts the sum of squared errors for all cases (Color figure online)



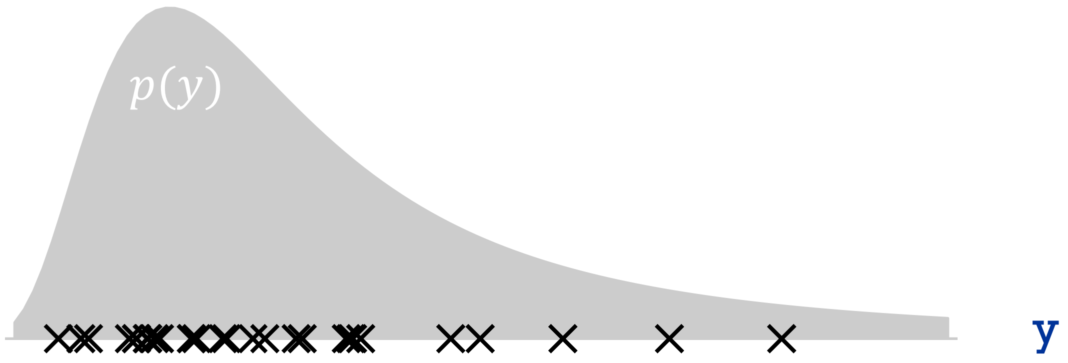
Cost-sensitive Learning Approaches

- Basic Idea: Alter optimization “cost” to reduce focus on the target’s mean
- Typical basic Technique: Weight the influence of each sample on the loss
- Pros:
  - Does not remove any information
  - Does not add noise to data
- Cons:
  - A method may not be directly applicable to all machine learning models

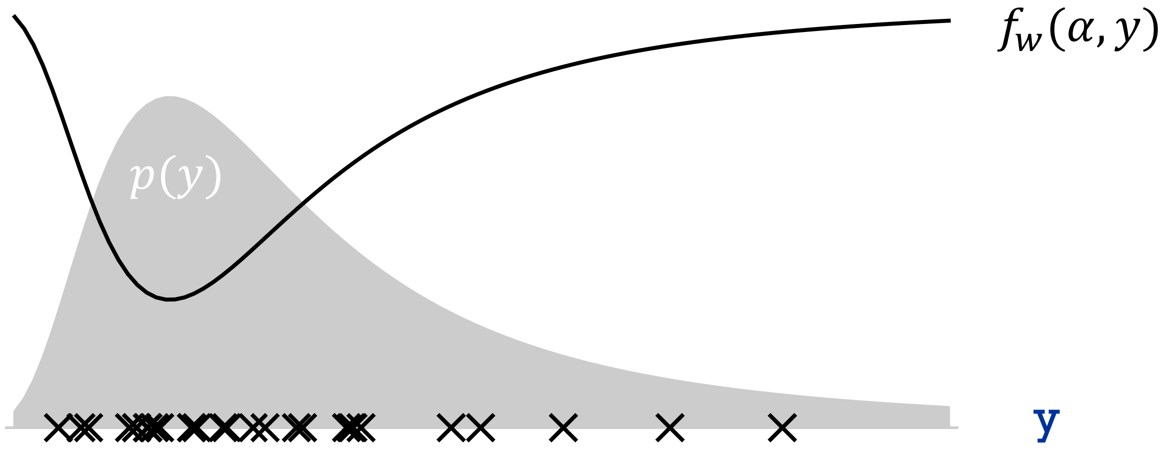
DenseWeight



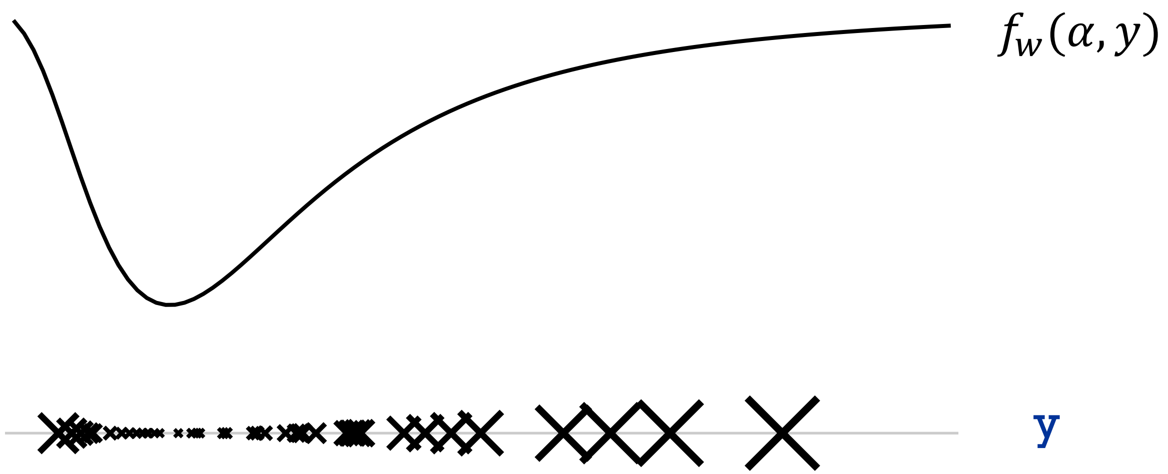
DenseWeight



DenseWeight



DenseWeight



Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

DenseWeight

data science

Desirable properties for f_w:

- Larger weights for rare data points in comparison to common data points
- Control extent of density-based weighting with a parameter α
- **No negative weights**
- **No 0 weights** for data-points (regardless of scaling) to not ignore parts of the dataset completely
- **Mean weight** over all training data points **of 1**

DenseWeight

Scaling factor

Inversion

Normalized density

Cut-off value

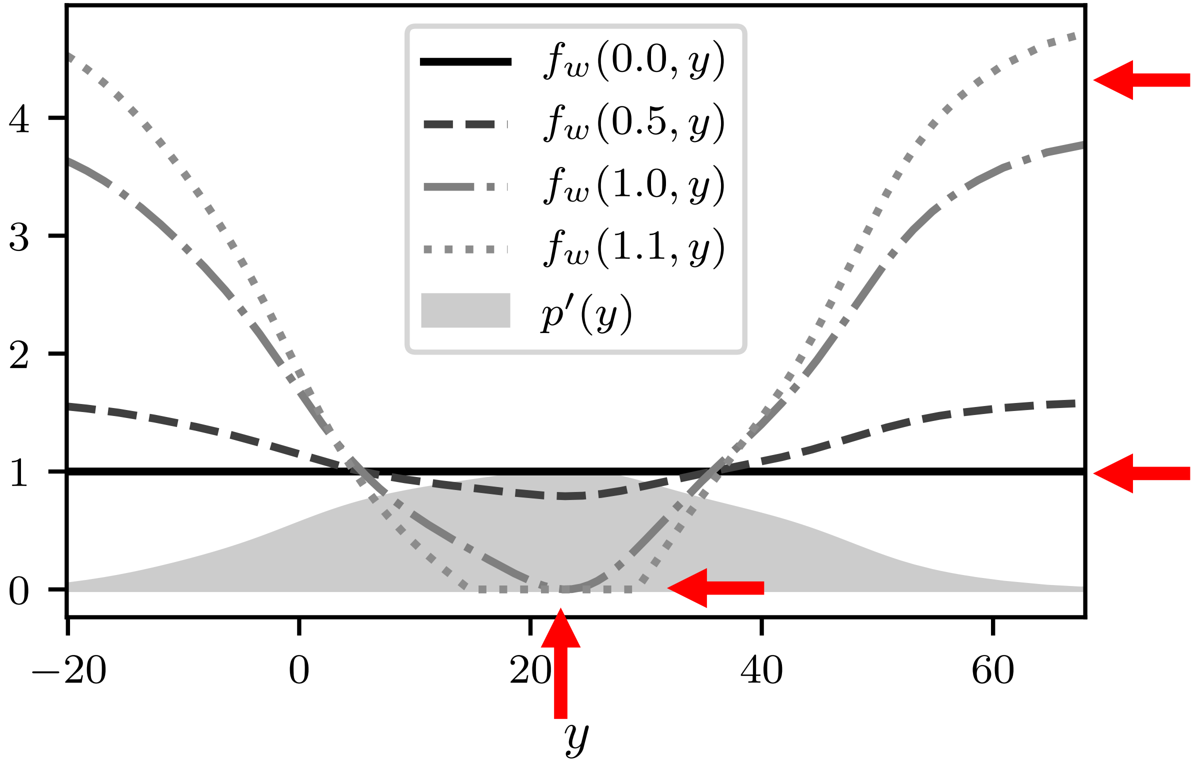
$$f_w(\alpha, y) = \frac{\max(\boxed{1 -}\,\boxed{\alpha}\,\boxed{p'(y)},\boxed{\epsilon})}{\boxed{\frac{1}{N}\sum_{i=1}^{N}(\max(1 - \alpha p'(y_i), \epsilon))}}$$

Normalization

DenseWeight

$$f_w(\alpha, y) = \frac{\max(1 - \alpha p'(y), \epsilon)}{\frac{1}{N}\sum_{i=1}^{N}(\max(1 - \alpha p'(y_i), \epsilon))}$$

- $\alpha \in [0, \infty[$ controls how much the model pays attention to **rare** data points vs. **common** data points
  - Larger $\alpha \rightarrow$ more attention to rare data points



DenseLoss

- Combining **DenseWeight** and **sample weighting for loss functions:**

→ **Cost-sensitive** approach for imbalanced regression called **DenseLoss:**

$$L_{DenseLoss}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \boxed{f_w(\alpha, y_i)} * \boxed{M(\hat{y}_i, y_i)}$$

Experiments

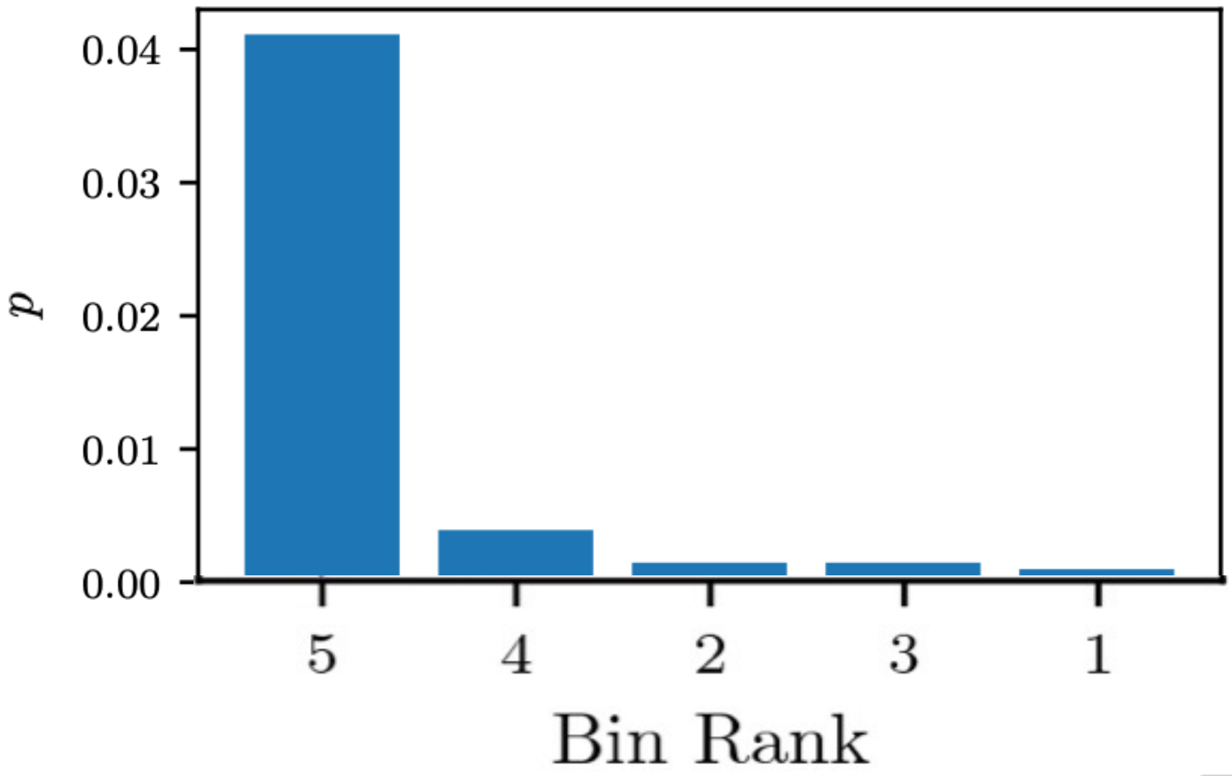
- We conduct experiments with
  - 4 synthetic toy datasets
  - 20 datasets used in the SMOGN paper
  - 1large precipitation dataset (Statistical downscaling of precipitation)

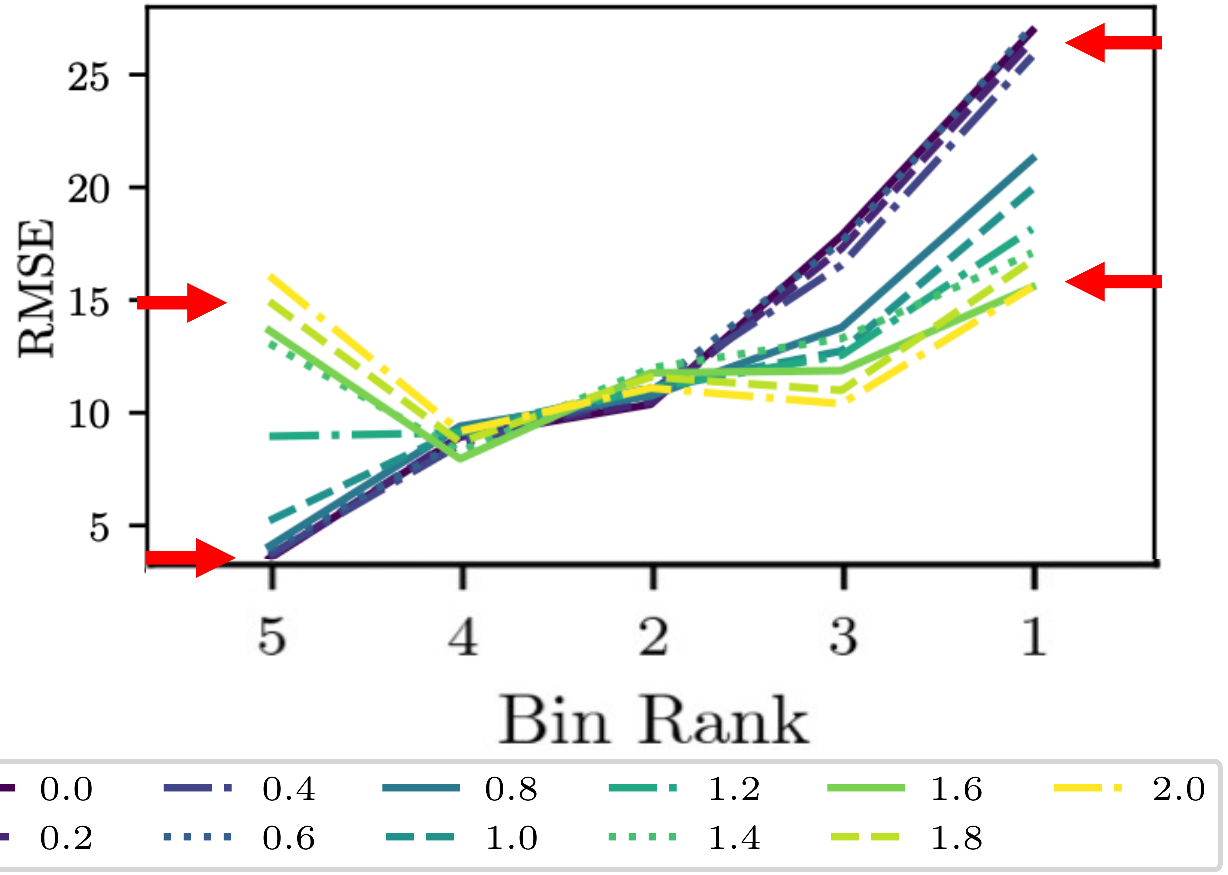
- $\varepsilon = 10^{-6}$
- α varies depending on the experiment

Experimental Setup (Synthetic datasets)

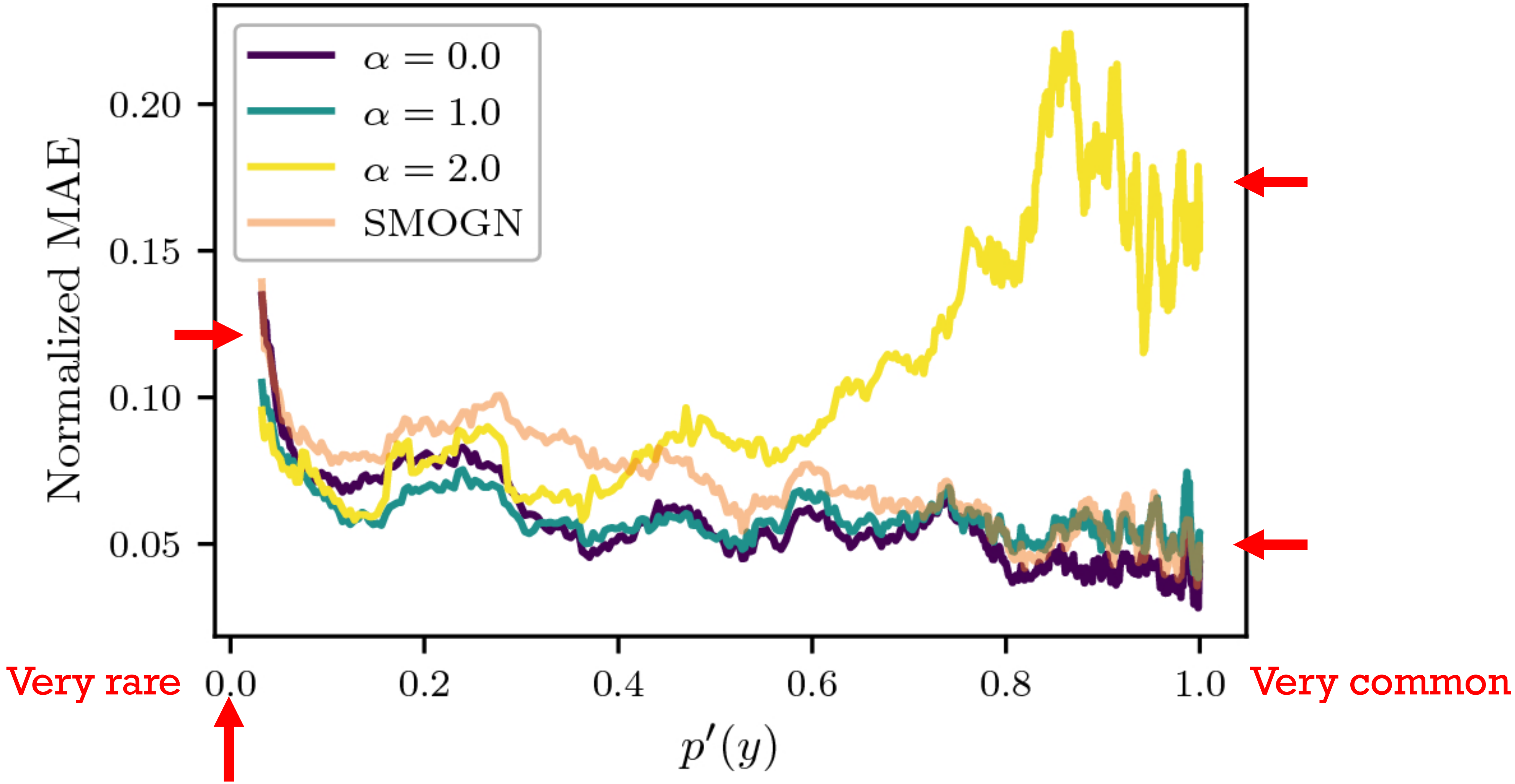
- Model:
  - Multi-layer Perceptron with 3 ReLU-activated hidden layers and 10 neurons each
  - DenseLoss with mean squared error (MSE)
  - Adam optimizer with learning rate 10^{-4} and weight decay coefficient 10^{-9}
  - Train for at most 1000 epochs with early stopping and a patience of 10 epochs
- Dataset splits: 60 % training, 20 % validation, 20 % test
- Evaluate α from 0.0 to 2.0 in steps of 0.2
- Train 20 model instances per α to assure reliability of results

Results (Synthetic dataset pareto)





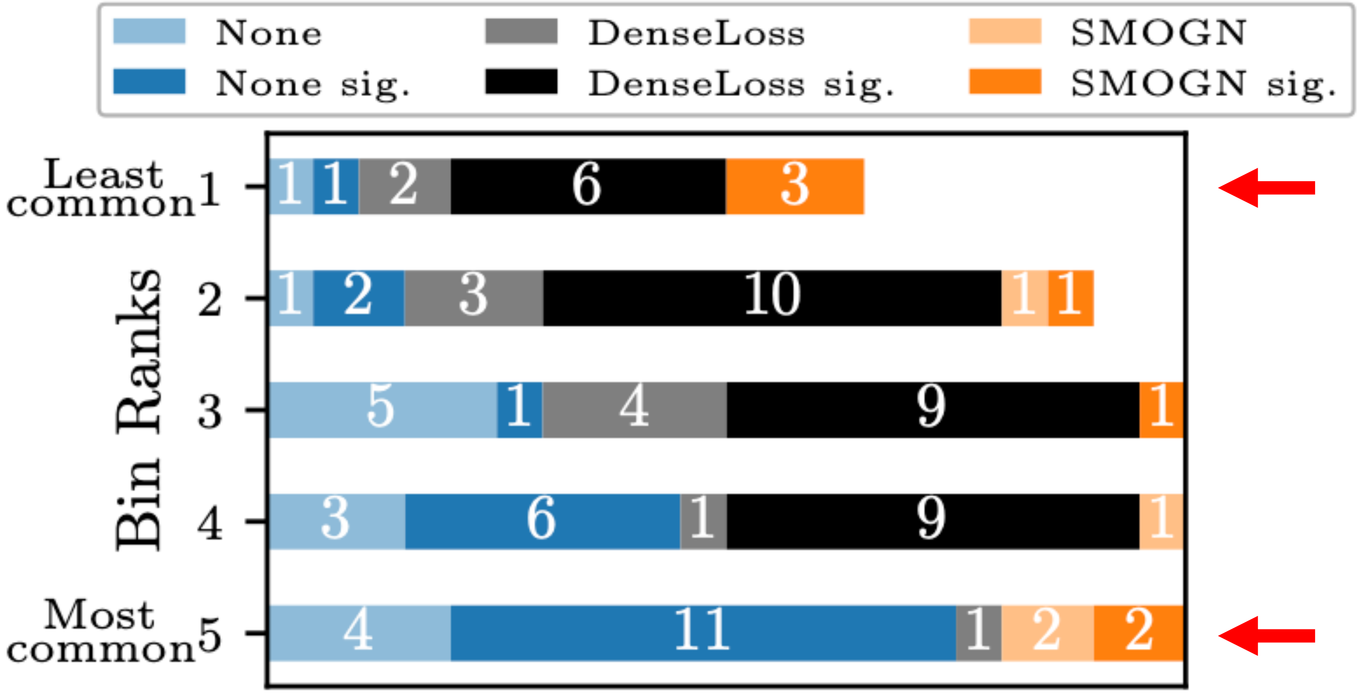
Results (Synthetic datasets)



Experimental Setup (SMOGN datasets)

- Model: same as for the synthetic datasets
- Dataset splits: 60 % training, 20 % validation, 20 % test
- Set $\alpha = 1.0$ for DenseLoss
- SMOGN is run with the same hyperparameters that the original authors used on these datasets
- Train 20 model instances per method to assure reliability of results

Results (SMOGN datasets)

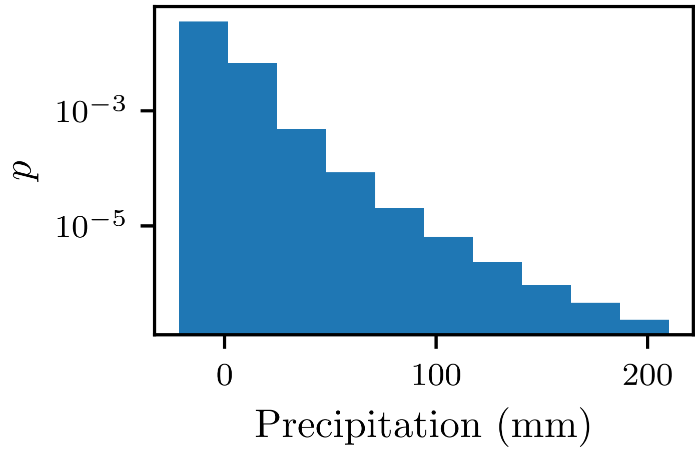


Number of datasets won per method for each bin rank

- DenseLoss most often provides lowest RMSE in the rarest bins
- As expected: Applying no method for imbalanced regression typically provides best performance in the most common bin

Experimental Setup (Precipitation dataset)

- Model: DeepSD[1]
- Task: Improve resolution of daily precipitation data from 128 km to 64 km
- Dataset splits: 1981-2005 training set, 2006-2014 test set
- Evaluate α from 0.0 to 4.0 in steps of 0.2
- Train 20 model instances per alpha to assure reliability of results



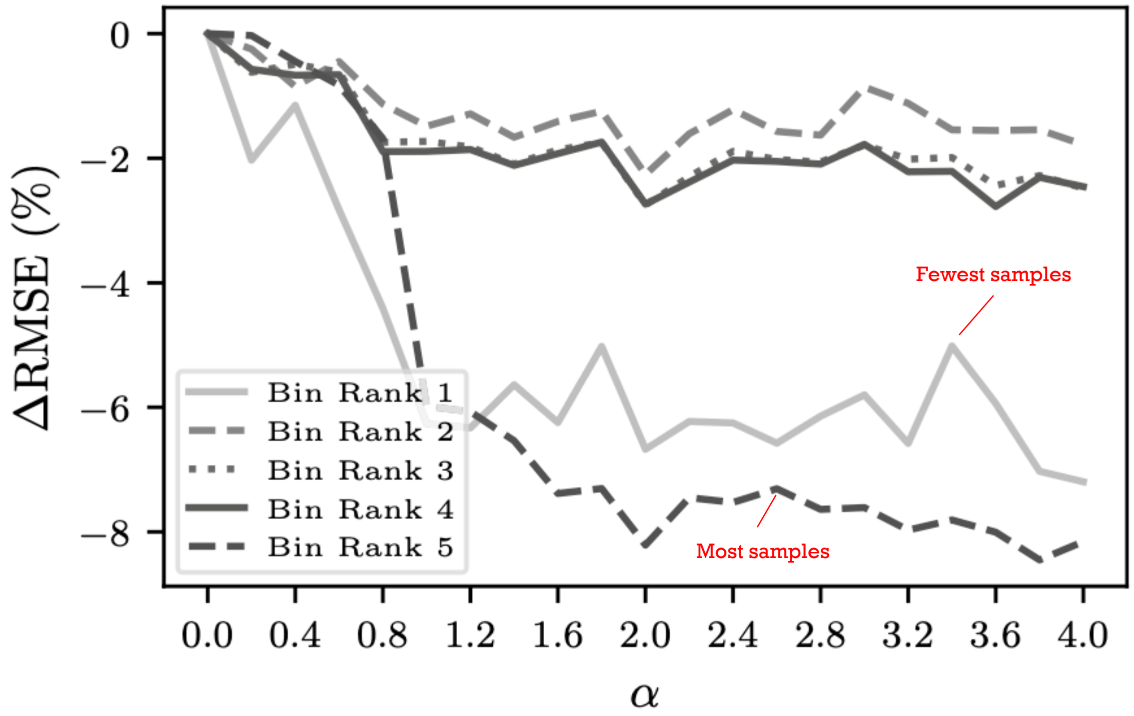
[1] Vandal, Thomas, et al. "Deepsd: Generating high resolution climate change projections through single image super-resolution." KDD 2017.

Results (Precipitation dataset)

- RMSE improved the most in both
  - the bin with the most samples (Bin Rank 5)
  - the bin with the fewest samples (Bin Rank 1)

→DenseLoss improves performance for rare samples for Deep Learning models

→Performance for very common samples also improved in this case



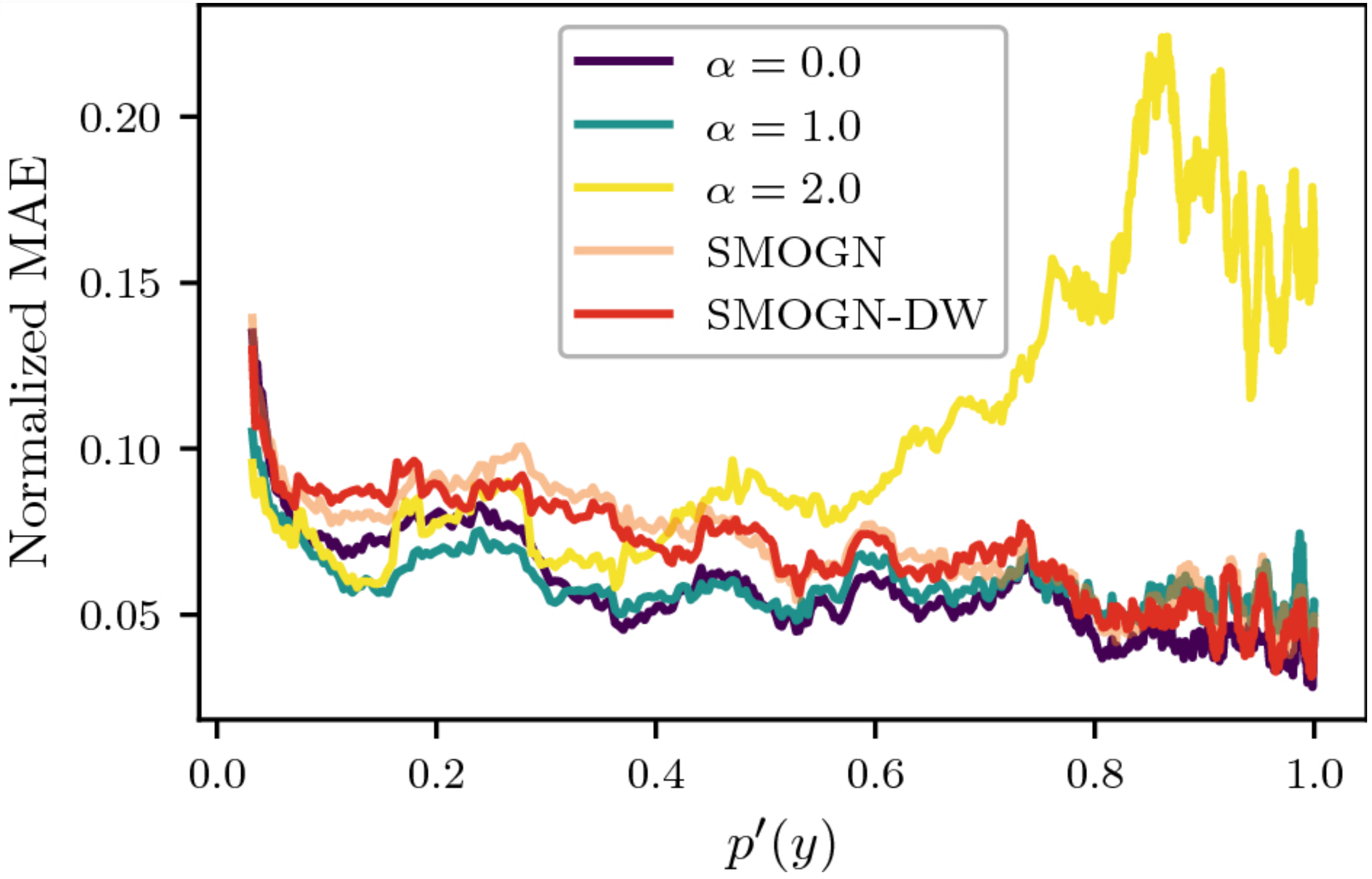
SMOGN with DenseWeight

- Observation: DenseLoss typically outperforms SMOGN
- Is the performance difference due to the **different measures of rarity** or due to the **difference between resampling and cost-sensitive learning?**

➢ Adapt SMOGN to use DenseWeight as its relevance function (SMOGN-DW)

- Experimental Setup:
  - $\alpha = 1$ for SMOGN-DW
  - Same experiments as before

SMOGN with DenseWeight



- ➢ Performance difference mostly due to resampling

Conclusion

- We propose our sample weighting approach for imbalanced regression **DenseWeight** and our cost-sensitive learning method **DenseLoss** based on DenseWeight
- Actively decide on the trade-off between focusing on common or rare cases through a single hyperparameter

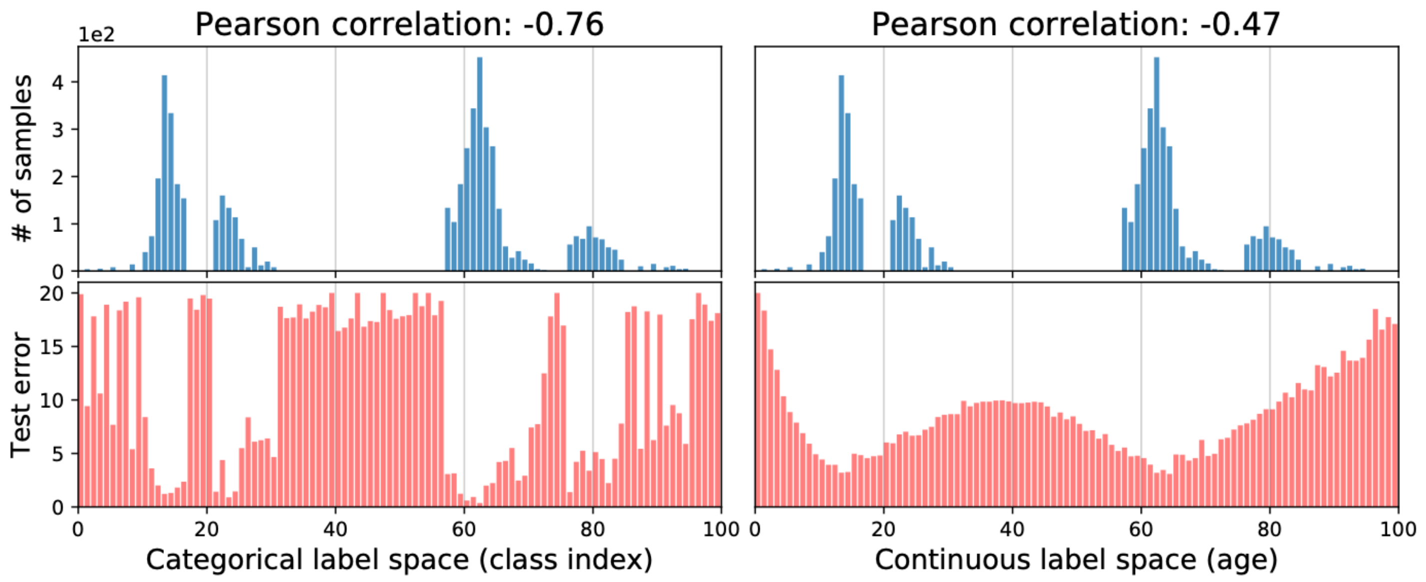
- Our approach can improve model performance for rare data points
- DenseLoss typically outperforms the sampling-based method SMOGN
- DenseLoss can be successfully applied for Deep Learning models

Code is available at: https://github.com/SteiMi/denseweight

Label Distribution Smoothing & Feature Distribution Smoothing

- From: Delving into Deep Imbalanced Regression. Y. Yang, K. Zha, Y. Chen, H. Wang, and D. Katabi. *Proceedings of the 38th International Conference on Machine Learning , volume 139 of Proceedings of Machine Learning Research, page 11842--11851. PMLR, (18--24 Jul 2021)*

Label Distribution Smoothing & Feature Distribution Smoothing



(a) CIFAR-100 (subsampled) (b) IMDB-WIKI (subsampled)

Figure 2. Comparison on the test error distribution (bottom) using same training label distribution (top) on two different datasets: (a) CIFAR-100, a classification task with categorical label space. (b) IMDB-WIKI, a regression task with continuous label space.

[Yang et al. 2021]

Label Distribution Smoothing (LDS)

- Also uses KDE to learn the effective imbalance in a dataset
- Convolve a symmetric kernel with the empirical density distribution

→ effective label density distribution: $\tilde{p}(y') \triangleq \int_{\mathcal{Y}} \mathrm{k}(y, y')p(y)dy$

- This can be used for cost-sensitive reweighting, e.g.
  - Inverse weighting

- Similar to DenseWeight/DenseLoss but without scaling and normalization

[Yang et al. 2021]

Label Distribution Smoothing (LDS)

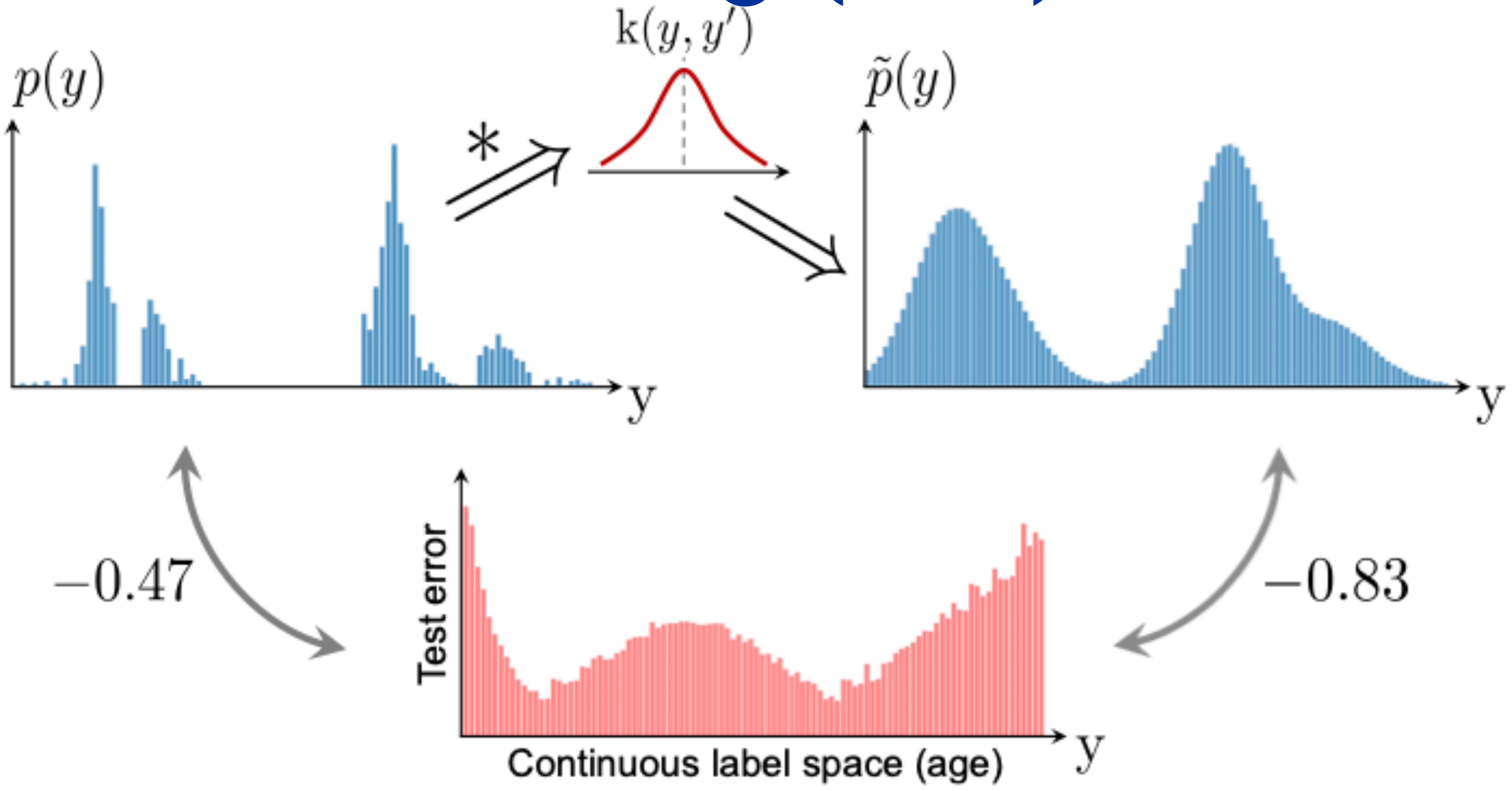


Figure 3. Label distribution smoothing (LDS) convolves a symmetric kernel with the empirical label density to estimate the effective label density distribution that accounts for the continuity of labels. [Yang et al. 2021]

Feature Distribution Smoothing (FDS)

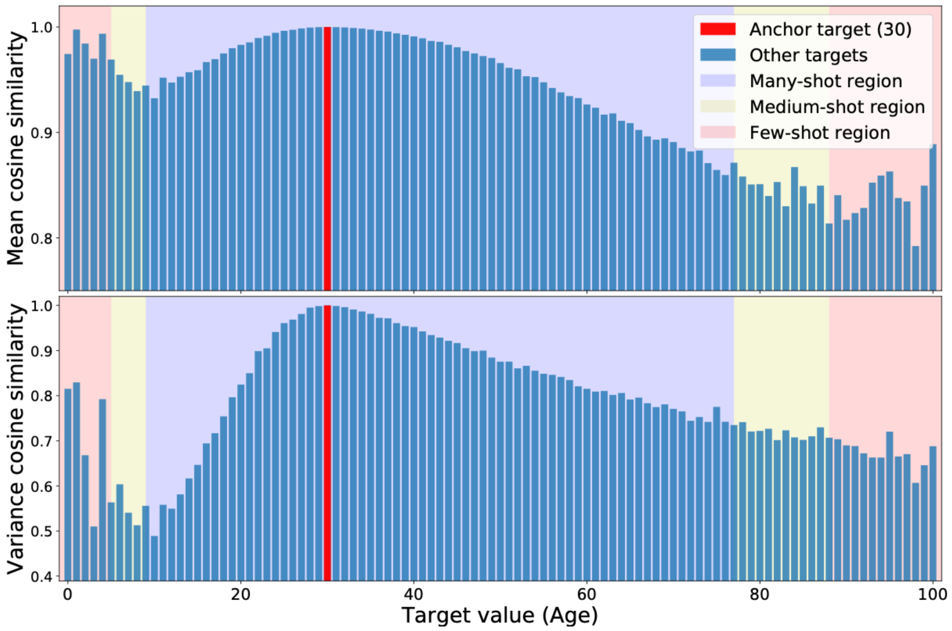


Figure 4. Feature statistics similarity for age 30. **Top:** Cosine similarity of the feature mean at a particular age w.r.t. its value at the anchor age. **Bottom:** Cosine similarity of the feature variance at a particular age w.r.t. its value at the anchor age. The color of the background refers to the data density in a particular target range. The figure shows that nearby ages have close similarities; However, it also shows that there is unjustified similarity between images at ages 0 to 6 and age 30, due to data imbalance.

[Yang et al. 2021]

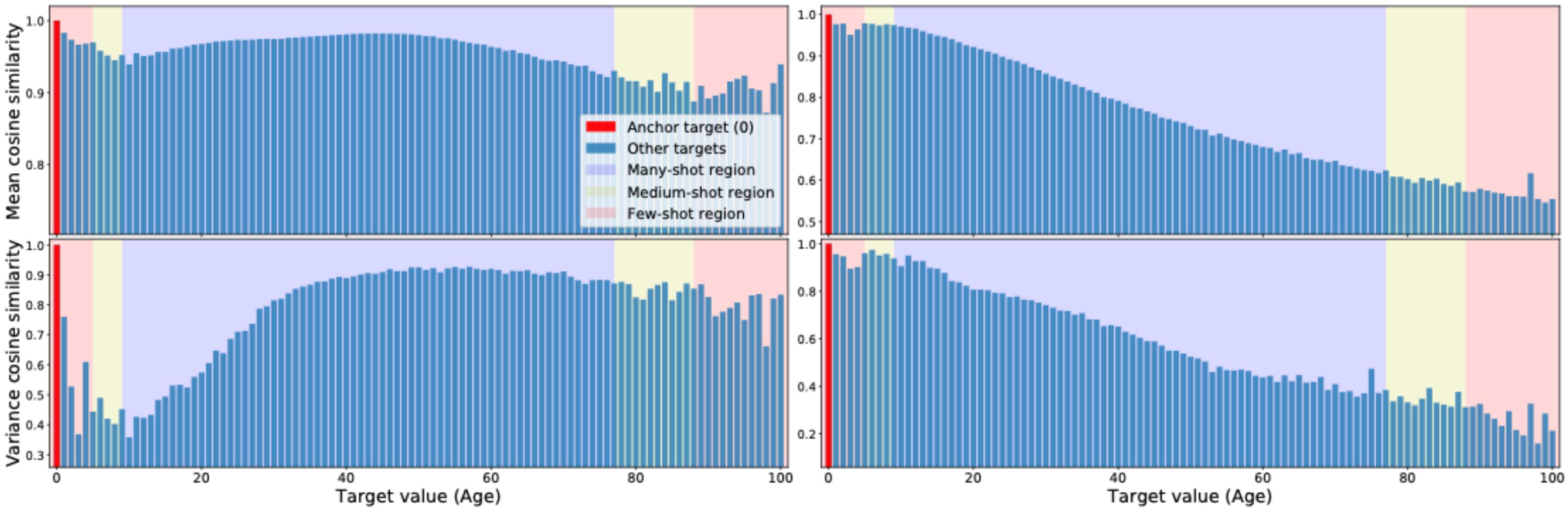
Feature Distribution Smoothing (FDS)

- Intuition: continuity in the target space → continuity in the feature space

- Steps:
  - Bin the target value range
  - Calculate mean and covariance of the model's learned features per bin
  - Smooth means and covariances by convolving a symmetric kernel over the bins
  - Calibrate features for each input sample with the help of the smoothed statistics
- Implementation: Feature calibration layer after the final feature map

[Yang et al. 2021]

Feature Distribution Smoothing (FDS)



(a) Feature statistics similarity for age 0, without FDS (b) Feature statistics similarity for age 0, with FDS

[Yang et al. 2021]

Feature Distribution Smoothing (FDS)

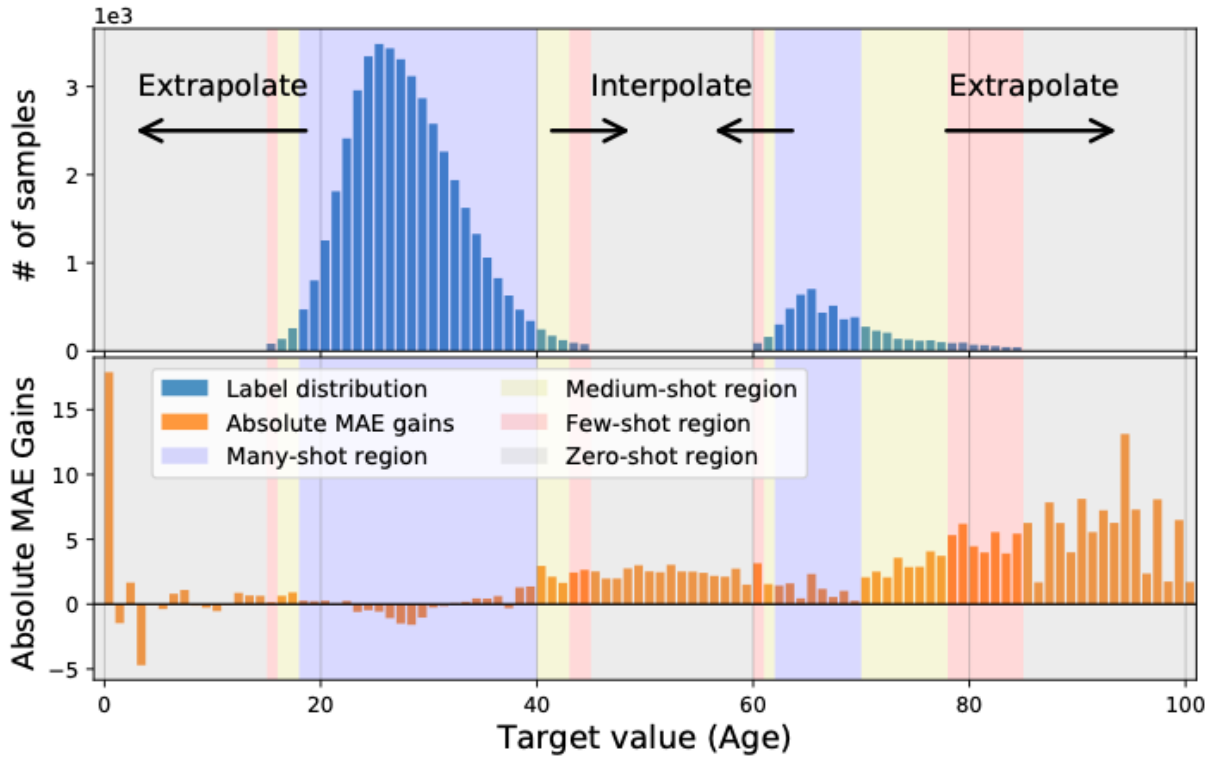


Figure 7. The absolute MAE gains of LDS + FDS over the vanilla model, on a curated subset of IMDB-WIKI-DIR with certain target values having no training data. We establish notable performance gains w.r.t. all regions, especially for extrapolation & interpolation.

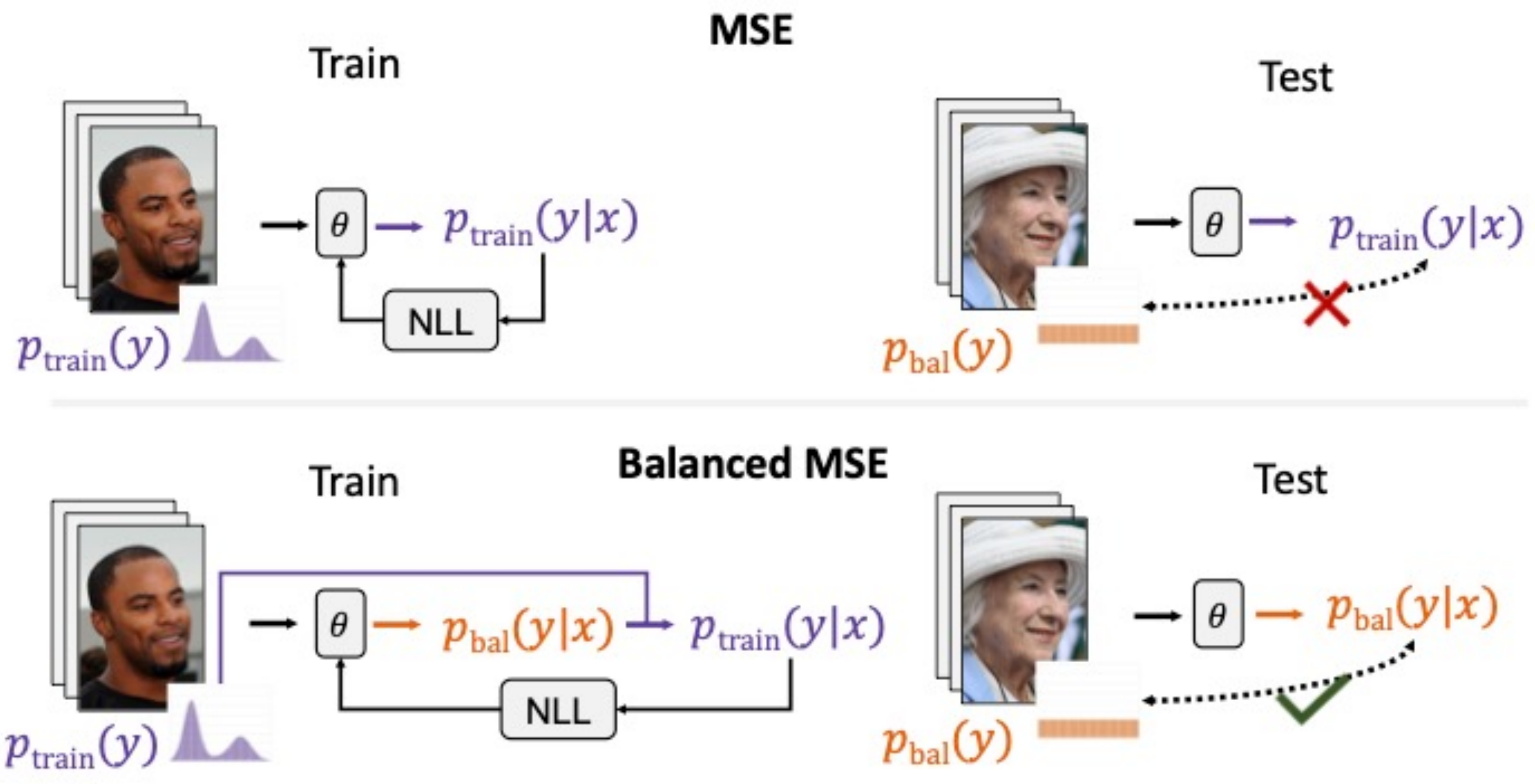
[Yang et al. 2021]

Balanced MSE
[Ren et al. 2022]

- Very recent work (March 30th 2022 on Arxiv)

- Propose a novel Mean Square Error (MSE) variant called **Balanced MSE**
- Focus on imbalanced visual regression
  - Age estimation
  - Pose estimation

Balanced MSE
[Ren et al. 2022]



Balanced MSE
[Ren et al. 2022]

MSE

$$L \cong -\log \mathcal{N}(\boldsymbol{y}; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I})$$

$$+ \log \int_Y \mathcal{N}(\boldsymbol{y}'; \boldsymbol{y}_{\text{pred}}, \sigma^2_{\text{noise}}\mathrm{I}) \cdot p_{\text{train}}(\boldsymbol{y}') d\boldsymbol{y}'$$

Balancing term

- Caveat: Integral can be difficult to compute
- Solutions: Closed-form with Gaussian Mixture Model or numerical, e.g., with Monte-Carlo

Balanced MSE
[Ren et al. 2022]

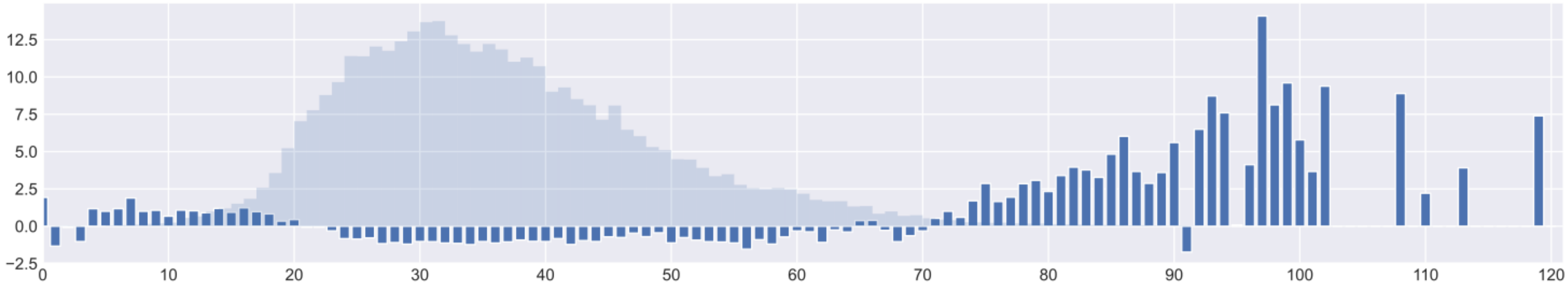


Figure 5. Balanced MSE's bMAE gain over the baseline. The light blue area in the background shows the training label histogram of IMDB-WIKI-DIR. Balanced MSE improves the performance on tail labels (age $<$ 20 and $>$ 70) substantially.

Conclusion

- **Resampling** approaches **SMOTE for regression, SMOGN**, and **Geometric SMOTE** provide easy-to-use algorithm-independent pre-processing techniques
- The metric **SERA** can be used to evaluate how well models estimate rare and common samples
- Several new **algorithm-level/cost-sensitive approaches** now available:
    - **DenseWeight/DenseLoss**
    - **Label/Feature Distribution Smoothing**
    - **Balanced MSE**
- Imbalanced regression, after years of "neglect", nowadays a very active research topic

Thank you for your attention!

References

- Branco, P., Torgo, L., & Ribeiro, R. P. (2017, October). SMOGN: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications* (pp. 36-50). PMLR.
- Camacho, L., Douzas, G., & Bacao, F. (2022). Geometric SMOTE for regression. *Expert Systems with Applications*, 116387.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.
- Douzas, G., & Bacao, F. (2019). Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information sciences*, *501*, 118-135.
- He, H., Bai, Y., Garcia, E. A., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322-1328). IEEE.
- Ren, J., Zhang, M., Yu, C., & Liu, Z. (2022). Balanced MSE for Imbalanced Visual Regression. *arXiv preprint arXiv:2203.16427*.
- Ribeiro, R.P., Moniz, N. Imbalanced regression and extreme value prediction. *Mach Learn* **109**, 1803–1835 (2020). https://doi.org/10.1007/s10994-020-05900-9
- Steininger, M., Kobs, K., Davidson, P. *et al.* Density-based weighting for imbalanced regression. *Mach Learn* **110**, 2187–2211 (2021). https://doi.org/10.1007/s10994-021-06023-5
- Torgo, L., Ribeiro, R. P., Pfahringer, B., & Branco, P. (2013, September). Smote for regression. In *Portuguese conference on artificial intelligence* (pp. 378-389). Springer, Berlin, Heidelberg.
- Yang, Y., Zha, K., Chen, Y., Wang, H., & Katabi, D. (2021, July). Delving into deep imbalanced regression. In *International Conference on Machine Learning* (pp. 11842-11851). PMLR.