# Smoothed Bootstrap et génération de données synthétiques pour la modélisation des extrêmes

Samuel Stocksieker [1,2] & Denys Pommeret [2] & Arthur Charpentier [3]

[1] *Université Claude Bernard Lyon 1, Laboratoire de Sciences Actuarielle et Financière (UR SAF), Lyon, 69007, France, samuel.stocksieker@univ-amu.fr*
[2] *Aix Marseille Univ, CNRS, I2M, Marseille, France, denys.pommeret@univ-amu.fr*
[3] *Université du Québec à Montréal, Canada, charpentier.arthur@uqam.ca*

**Résumé.** En apprentissage supervisé, il est assez fréquent de se retrouver confronté à des données présentant des distributions déséquilibrées. Cette situation entraîne souvent une difficulté d'apprentissage pour les algorithmes standards. La recherche et les solutions en matière d'apprentissage à partir de distributions déséquilibrées se sont principalement concentrées sur les tâches de classification. Malgré son importance, très peu de solutions existent pour la régression déséquilibrée (*Imbalanced Regression*). Dans cet article, nous proposons une procédure d'augmentation de données, nommée DENIS, basée sur des estimations à noyau de densité. Cette approche fournit une expression des densités conditionnelles des générateurs. Nous appliquons DENIS en régression déséquilibrée et proposons de le combiner à une nouveau type de générateur de type wild-boostrap pour simuler la variable cible, conditionnellement aux nouvelles données synthétiques. Nous évaluons les performances de l'algorithme DENIS dans des situations de régression déséquilibrée. Nous évaluons empiriquement et comparons notre approche et démontrons une amélioration significative par rapport aux techniques existantes.

**Abstract.** In supervised learning, it is quite common to encounter data with imbalanced distributions. This situation often leads to learning difficulties for standard algorithms. Research and solutions in imbalanced learning have mainly focused on classification tasks. Despite its importance, very few solutions exist for imbalanced regression. In this paper, we propose a data augmentation procedure, called DENIS, based on kernel density estimates. This approach provides an expression of the conditional densities of the generators. We apply DENIS in imbalanced regression by combining such generation procedures with a bootstrap resampling technique for the target values. We evaluate the performance of the DENIS algorithm in imbalanced regression situations. We empirically evaluate and compare our approach and demonstrate significant improvement over existing techniques.

**Keywords.** Smoothed Bootstrap, Kernel Density Estimate, Imbalanced, Synthetic Data

# 1 Introduction

Many real-world forecasting problems are based on predictive models in a supervised learning framework and the standard algorithms fail when the target variable is skewed. The learning from imbalanced data concerns many problems with numerous applications in different fields

[27], [21]. The major part of such works concerns imbalanced classification (see for instance [7] or [9] or [13] or [26] or [44]. As shown by [4], many solutions for dealing with imbalanced learning propose a pre-processing strategy especially the generation of new synthetic data. A large part of these existing methods consists of combining the well-known SMOTE algorithm [22].

In the literature, the imbalanced regression corresponds to *the correct prediction of rare extreme values of a continuous target variable* [22] but, contrary to the classification tasks, there is no level to quantify the imbalance and the labels are continuous. Unlike in a classification context, learning from an imbalanced dataset for regression tasks leads to two additional problems: i) the definition of the imbalanced phenomenon and ii) the identification of the observations that are considered as minority. Regression tasks over imbalanced data are not as well explored. Few works have addressed the problem despite the importance of this topic. The first and main works on this topic propose to binarize the problem with a relevant function and an associate threshold [40] in order to adapt some Imbalanced classification solutions [41], [5], [6], [31], [37], [8] for instance. This methodology presents the disadvantage of dividing the continuous distribution of the target variable into classes and therefore involves a loss of information. More recently, new other methods have emerged by using deep learning approaches such as [33], [16], or [23]. [45] proposes to use kernel density estimates to improve learning from imbalanced data with continuous targets.

This paper aims to propose a novel approach, which we shall call DENIS (for Data Enrichment for Numerical Imbalanced Situations) to deal with the imbalanced regression problem. The first step of DENIS consists of generating synthetic data for covariates based on kernel density estimators. This data augmentation procedure can be used independently to generate synthetic data (supervised or non-supervised). A second step of DENIS is concerned with the imbalanced regression where the generator model is combined with a wild-bootstrap procedure to generate target values given the synthetic covariates. The DENIS algorithm can be easily used in an imbalanced classification where the label of the target variable remains unchanged. The main contributions of our paper can be summarized as follows: i) Propose a global form to group some existing perturbation generators; ii) Deducing new synthetic data oversampling methods; iii) when combined with a wild-bootstrap the generator model provides a new method for dealing with imbalanced regression iv) we empirically compared our generalized algorithm and its variants with several state-of-the-art approaches and we obtained great performance on several datasets.

The paper is organized as follows. In Section 2 we give a general form of our data augmentation procedure corresponding to the first step of DENIS. We study two standard perturbation methods that are included in this approach. In Section 3 we develop the theory to obtain new generators and we will look more closely at the imbalanced regression. This is the second step of DENIS where we combine generators with a wild bootstrap to generate synthetic target values. Numerical results on several applications are presented in Section 5.

# 2 A New Kernel-Based Oversampling Approach

## 2.1 General Formulation

We consider a sequence of observations $\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$, which are realizations of $n$ iid random variables $(\boldsymbol{X}, Y)$, where the target variable $Y$ is univariate and the covariate $\boldsymbol{X}$ is a $p$-dimensional random vector. The components of $\boldsymbol{X} = (X_1, \cdots, X_p)$ are supposed to be continuous or discrete and $Y$ is supposed to be either qualitative (classification) or quantitative (regression). Write $\tilde{\boldsymbol{x}} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ the set of all observations. We propose a generalized oversampling procedure based on the form of the following weighted kernel density estimate:

$$g_{\boldsymbol{X}^*}(\boldsymbol{x}^*|\tilde{\boldsymbol{x}}) = \sum_{i \in \mathcal{I}} \omega_i K_i(\boldsymbol{x}^*, \tilde{\boldsymbol{x}}), \tag{1}$$

where $(K_i)_{i\in\mathcal{I}}$ is a collection of kernels, $(\omega_i)_{i\in\mathcal{I}}$ is a sequence of positive weights with $\sum_{i\in\mathcal{I}} \omega_i = 1$, and $\mathcal{I}$ represents a subset of $\{1, 2, \cdots, n\}$. Here the index $^*$ stands for the synthetic data. In (1) we propose a general form for the conditional density for the synthetic data generators. The objective is to use the flexibility of the kernels to estimate the density of covariates in order to obtain synthetic data that reflects the distribution of the observations.

We propose to show that (1) generalizes perturbation-based synthetic data oversampling. We give an illustration with the basic algorithms ROSE and Gaussian Noise in Sections 2.2 where we demonstrate that these methods are particular cases of the generalized form (1), with corresponding parameters. In Section 3 we will show that some new methods can be deduced from the generic form (1) and we will compare some of them to current competitors in the imbalanced regression context.

The generators in (1) can be considered as smoothed bootstrap methods ([35], [24], [14]). Indeed, the smoothed bootstrap consists in drawing samples from kernel density estimators of the distribution. This bootstrap can be decomposed into two steps: first, a seed is randomly drawn and second, a random noise from the kernel density estimator is added to obtain a new sample. In the form (1), the first step is represented by the drawing weight $\omega_i$ and the second by the kernel $K_i(x)$. Convergence properties of smoothed bootstrap are given in [15] and [20]. As described by the authors, the smoothed bootstrap *provides better performances than classical bootstrap when a proper choice of smoothing parameters is used*. They proved the consistency of the smoothed bootstrap with the classical multivariate kernel estimator and more specifically the convergence in Mallow's metric. Other works have focused on the consistency of the multivariate kernel density estimate and proposed a relevant bandwidth matrix, for instance, [34], [32] and [19]. Note also that a kernel density estimator is a special case of mixture models with as many components as observations.

## 2.2 Rewriting Perturbation Approaches

We illustrate (1) by recovering two classical data augmentation procedures as follows:

- At each step of the ROSE algorithm (see [30]) the seed $S$ is selected randomly. Given $S$

synthetic data is generated with a multivariate density

$$g^{ROSE}_{\boldsymbol{X}^*}(\boldsymbol{x}^*|\tilde{\boldsymbol{x}}, S=i) = K^{ROSE}_{H_n}(\boldsymbol{x}^* - \boldsymbol{x}_i) = \frac{1}{|H_n|^{1/2}} K(H_n^{-1/2}(\boldsymbol{x}^* - \boldsymbol{x}_i))),$$

where $K$ denotes the multivariate Gaussian kernel and $H_n = diag(h_1, ..., h_p)$ is the bandwidth matrix proposed by Bowman and Azzalini [2], with $h_q = (\frac{4}{(p+2)n})^{1/(p+4)}\widehat{\sigma}_q, q = 1, ..., p$. Finally, a synthetic random variable $\mathbf{X}^*$ is generated with the density

$$g^{ROSE}_{\boldsymbol{X}^*}(\boldsymbol{x}^*|\tilde{\boldsymbol{x}}) = \frac{1}{n}\sum_{i=1}^{n} K^{ROSE}_{H_n}(\boldsymbol{x}^* - \boldsymbol{x}_i) = \sum_{i=1}^{n} \omega_i K^{ROSE}_{H_n}(\boldsymbol{x}^* - \boldsymbol{x}_i)).$$

- Similarly to ROSE, at each step of the Gaussian Noise algorithm (see [29]) a seed is selected and synthetic data is generated. Finally, the generating multivariate density has the form

$$g^{GN}_{\boldsymbol{X}^*}(\boldsymbol{x}^*|\tilde{\boldsymbol{x}}) = \frac{1}{n}\sum_{i=1}^{n} K^{GN}_{H_n}(\boldsymbol{x}^* - \boldsymbol{x}_i) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|H_n|^{1/2}} K(H_n^{-1/2}(\boldsymbol{x} - \boldsymbol{x}_i)),$$

where $H^{GN}_n = diag(h_1, ..., h_p)$, $h_q = \sigma_{noise}\widehat{\sigma}_q, q = 1, ..., p$.

Both cases are particular cases of (1) with $\omega_i = \frac{1}{n}$ and $K_i(\tilde{\boldsymbol{x}}, \boldsymbol{x}) = K_{H_n}(\boldsymbol{x} - \boldsymbol{x}_i)$, i.e. the same Gaussian kernel for all observations but with a different bandwidth matrix.

However, with these generators, the directions in the data space are randomly generated and so they can more explore the space. The distance between the new sample and the seed is also unbounded. However, the directions are randomly chosen and do not respect the correlation between the data and their support and the correlations between variables.

# 3 New Kernel-Based Methods

As the ROSE and GN techniques use a multivariate Gaussian kernel estimate with a diagonal bandwidth matrix, we can rewrite their associated generating density as follows:

$$g_{\boldsymbol{X}^*}(\boldsymbol{x}^*|\tilde{\boldsymbol{x}}) = \sum_{i=1}^{n} \omega_i \prod_{j=1}^{p} K_{h_j}(x^*_j - x_{ij}) \tag{2}$$

with $K_{h_j}(u) = (2\pi)^{-1/2} h_j^{-1} e^{-\frac{1}{2{h_j}^2}u^2}$ the univariate gaussian kernel density estimator with smoothing parameter $h_j$. Such kernels are clearly not adapted for asymmetric, bounded or discrete variables. This remark is also true for the work of [45] which uses some symmetric kernels to improve the learning of imbalanced datasets. Another remark about this work is the division of the target variable support into $B$ groups that involve a loss of information.

To fix the drawback of the classical kernel we extend (2) by adapting (1) to the support of $\boldsymbol{x}$, considering some non-classical kernels (we refer to some works handling the kernel density

estimation for specific distributions inspired from [1], [36], [25], [12]). We suggest rewriting the form (1) as

$$g^{per}_{\boldsymbol{X}^*}(\boldsymbol{x}^*|\tilde{\boldsymbol{x}}) = \sum_{i\in\mathcal{I}} \omega_i \prod_{j=1}^{p} K_{h_j}(x^*_j, x_{ij})$$

where $K_{h_j}(u,x)$ is a univariate kernel adapted to the nature of the $j$th variable and specifically defined on $x$ as follows:

- Gaussian kernel for a variable defined on $\mathbb{R}$ (classical kernel):

$$K_h(u,x) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{u-x}{h})^2}.$$

- Binomial kernel for a discrete variable defined on $\mathbb{N}$:

$$K_h(u,x) = \frac{(x+1)!}{u!(x+1-u)!}\left(\frac{x+h}{x+1}\right)^u \left(\frac{1-h}{x+1}\right)^{x+1-u}.$$

- Gamma kernel for a positive asymmetric distribution defined on $[a,+\infty]$:

$$K_h(u,x) = \frac{u^{(x-a)/h}}{\Gamma(1+(x-a)/h)h^{1+(x-a)/h}} \exp\big(\frac{-u}{h}\big)\mathbb{1}_{[a,+\infty]}(u).$$

- Negative Gamma kernel for a negative asymmetric distribution defined on $[-\infty,b]$:

$$K_h(u,x) = \frac{u^{-(x-b)/h}}{\Gamma(1-(x-b)/h)h^{1-(x-b)/h}} \exp\big(\frac{-u}{h}\big)\mathbb{1}_{[-\infty,b]}(u).$$

- Beta kernel for a variable defined on $[0,1]$:

$$K_h(u,x) = \frac{u^{x/h}(1-u)^{(1-x)/h}}{\mathcal{B}(\frac{x}{h}+1, \frac{1-x}{h}+1)}\mathbb{1}_{[0,1]}(u).$$

- Truncated Gaussian kernel for a variable defined on $[a,b]$:

$$K_h(u,x) = \frac{\alpha}{h\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{u-x}{h})^2}\mathbb{1}_{[a,b]}(u), \alpha := \left(\int_a^b \frac{1}{h\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{u-x}{h})^2}\right)^{-1}$$

Note that if the Dirac kernel ($\mathbb{1}_{\boldsymbol{x}=\boldsymbol{x}_i}$) is used, we get the standard bootstrap: 1 includes also the simple oversampling. It is important to note that the DENIS algorithm uses an estimation of the smoothing parameter $h$ provided by some specific R-package dedicated to the density estimation (for instance, it uses the Silverman estimation for the Gaussian kernel). Their estimates are based on properties of univariate consistency. Another technique to deal with skewed or heavy-tailed distributions is to apply a transformation of the data in order to use classical kernel density estimation [10], [11] but it necessitates proposing a relevant transformation.

**Remark:** The use of a diagonal bandwidth matrix in (2) does not take into account the correlation between variables. To improve this issue, we could consider a full (symmetric positive definite) smoothing matrix. In that case, we would use a multivariate kernel density estimate considering the correlation between the variables which would be optimal for generating data. However, the estimation of this kind of matrix can be biaised because based on correlation matrix i.e. linear correlations. The thesis of Duong [17] and his associated R-package [18] proposed a Multivariate Gaussian Kernel but is a very time-expensive and offer a maximum of 6 dimensions. These works, despite their high quality, are very limited in practice since on the one hand datasets contain generally more than 6 variables, and on the other hand Gaussian Kernel estimators are inappropriate for mixed data.

# 4 DENIS as a Solution for Imbalanced Regression

Using the previous methods proposed we can generate synthetic covariates $X^*$. We then have to generate the target variable $Y$ given $X^*$ obtained from the first generator step of DENIS. In this sense we propose here to define the drawing weights $\omega_i$ as the inverse of the kernel density estimate for the target variable $Y$: the more isolated an observation is, the higher its drawing weight. The same idea is proposed in [38]. We also defined some safeguards to avoid getting some weights too high. Finally, the target variable is not generated in the same way as the covariate. Once the covariates are generated from our generator models, we propose to adapt a wild-bootstrap technique[1] [42] as follows: i) train a Random Forest on the initial sample; ii) predict target variable $\widehat{y_i}$; iii) draw uniformly a prediction error $\epsilon_k$ to generate a new $y_i^* := \widehat{y_i} + \epsilon_k v_i$ where $v_i$ is a random variable. We suggest an adaptation of this method to synthetic data in considering the impact of getting new covariate $y_i^* := \widehat{y_i} + |\epsilon_k| v_i \times sign(\widehat{y_i} - \widehat{y_i^*})$. This form is close to the Wild Bootstrap version with the Rademacher distribution. This is the second step of DENIS. (giving lower performances in our applications). The idea behind this proposition is to consider the prediction error and the impact of the synthetic covariate on the target variable. The choice of using a random forest is justified by its good predictive performance, its non-parametric nature, and the possibility of getting an error distribution for the same value of the target variable.

# 5 Application in Imbalanced Regression

Although the DENIS algorithm can be applied for the classification tasks, we focus on the imbalanced regression context because of the natural capacity of the form (1) to handle continuous variables. We test our approach on several real data set from a repository provided as a benchmark for imbalanced regression problems[2] and presented in [6].We compare our results to existing methods to deal with imbalanced regression from the *UBL* R-package ([3]): classical oversampling, SMOTE, Gaussian Noise, SMOGN, WERCS and ADASYN from the

[1]The kernel regression (Nadaraya-Watson estimator) was also tested but not selected because its high computation time and poor performance

[2]`https://paobranco.github.io/DataSets-IR/`

python-package *ImbalancedLearningRegression* ([43]). These techniques are used with their automatic relevance function and the same parameters as DENIS if any. To avoid sampling effects and obtain a distribution of prediction errors we ran 10 train-test datasets. In the same way, to avoid getting results dependent on some learning algorithms we use 10 models from the *autoML of the H2O R-package* [28] among the following algorithms: Distributed Random Forest, Extremely Randomized Trees, Generalized Linear Model with regularization, Gradient Boosting Model, Extreme Gradient Boosting and a Fully-connected multi-layer artificial neural network. It is also possible to use a clustering (Gaussian Mixture Model) in DENIS to apply a generation by cluster. Note that the ROSE algorithm did not exist for the imbalanced regression. We train, with the autoML, the following train dataset:

- Benchmark: UBL-Oversampling (*UBL-OS*), UBL-SMOTE for regression (*UBL-SMOTE*), UBL-Gaussian Noise for regression (*UBL-GN*), UBL-SMOGN for regression (*UBL-SMOGN*), UBL-WERCS (*UBL-WERCS*), IRL-ADASYN (*IRL-OS*)
- DENIS (step 1): Oversampling (*G-OS*), Gaussian Noise (*G-GN*), Gaussian Noise with GMM-clustering (*G-GNwCl*), ROSE (*G-ROSE*), ROSE with a GMM-clustering (*G-ROSEwCl*), Non classical Smoothed Boostrap with contraints on the distributions (*G-NCSB*), Classical Smoothed Boostrap (*G-CSB*).

Figure 1 presents RMSE gain (wrt the imbalanced dataset) and the median of the RMSE ranking. We can observe on these datasets that the DENIS algorithm empirically outperforms the state-of-the-art techniques, especially the Non-Classical Smoothed Bootstrap. The RMSE-rank represents the ranking of approaches according to the RMSE for a run: rank 1 corresponds to the training dataset that offers the smallest RMSE on the test sample. Beyond performance in prediction, non-classical kernels allow for generating more coherent synthetic data by preserving the domains of definition of the variables.

| RMSE gain | NO2 | cpuSm | Boston | Bank8FM | Abalone |
|---|---|---|---|---|---|
| UBL-OS | -3% | 2% | -7% | 63% | 0% |
| UBL-SMOTE | -10% | -2% | -12% | 27% | -4% |
| UBL-GN | -8% | -5% | -7% | 57% | -3% |
| UBL-SMOGN | -10% | -3% | -10% | 29% | -3% |
| UBL-WERCS | -4% | 3% | -1% | 57% | -1% |
| IRL-ADASYN | 10% | 22% | -1% | 69% | NA |
| G-OS | -1% | -2% | -3% | 57% | -3% |
| G-GN | -11% | -18% | -21% | 0% | -9% |
| G-GNwCl | -10% | -18% | -14% | 13% | -6% |
| G-ROSE | -9% | -17% | -17% | -6% | -9% |
| G-ROSEwCl | -9% | -19% | -21% | 0% | -8% |
| G-NCSB | -9% | -23% | -23% | -6% | -9% |
| G-CSB | -12% | -20% | -17% | 0% | -9% |

(a) RMSE gain

| RMSE rank | NO2 | cpuSm | Boston | Bank8FM | Abalone |
|---|---|---|---|---|---|
| Imb | 16,0 | 13,0 | 16,5 | 7,0 | 16,5 |
| UBL-OS | 14,0 | 14,5 | 13,0 | 17,0 | 16,5 |
| UBL-SMOTE | 10,0 | 11,5 | 11,0 | 11,0 | 12,0 |
| UBL-GN | 10,5 | 9,5 | 11,5 | 15,5 | 13,0 |
| UBL-SMOGN | 8,0 | 12,0 | 11,5 | 12,0 | 12,0 |
| UBL-WERCS | 14,0 | 16,0 | 15,5 | 16,5 | 15,0 |
| IRL-ADASYN | 18,0 | 18,0 | 16,0 | 18,0 | NA |
| G-OS | 16,0 | 14,0 | 14,5 | 15,5 | 13,5 |
| G-GN | 6,5 | 5,5 | 2,5 | 6,0 | 6,5 |
| G-GNwCl | 8,5 | 5,0 | 6,5 | 8,0 | 9,5 |
| G-ROSE | 8,5 | 4,0 | 4,0 | 4,5 | 7,0 |
| G-ROSEwCl | 10,0 | 4,0 | 3,0 | 8,0 | 8,0 |
| G-NCSB | 8,0 | 2,0 | 4,0 | 5,0 | 5,5 |
| G-CSB | 6,5 | 3,5 | 6,0 | 5,0 | 6,0 |

(b) Median of the RMSE-rank

Figure 1: RMSE-gain and median of the RMSE-rank on the Imbalanced Regression Datasets

We can see on these several applications, with several runs, several learning algorithms, and several performance metrics that the DENIS approach seems relevant to deal with imbalanced regression.

# 6 Discussion and Perspectives

DENIS is a new approach that offers a general form for the generator which is based both on the theoretical foundations of kernel estimators and classical smoothed bootstrap techniques. It provides a general expression for the conditional density of the generator. The use of well-chosen kernels makes it possible to take into account the nature of the covariates: continuous, discrete, totally or partially bounded. Numerical applications in imbalanced regression models demonstrate that DENIS and its variants are very competitive.

The weights $\omega_i$ offer a large flexibility to use the method. For instance, it is possible to handle classification tasks by conditioning with the minority class. We could deal with multi-class classification too. It is also possible to combine another weighting, proposed in the literature, to focus on specific samples in the synthetic data generation with a kernel approach to perform the methodology.

As a perspective, a natural extension of this work is to automate the choice of the kernel estimators, the weights, as well as some parameters according to the data. It will be possible to define a kernel according to the neighborhood in the same dataset. We also could define $\omega_i$ in order to generate to a target distribution as done in [39]. Finally, by nature, the perturbation approaches with a diagonal bandwidth matrix do not consider the correlation between covariates. The interpolation approaches consider it but the generation is limited to the segments. Another direction for further investigations would be to better consider the correlations between variables and be able to handle mixed data.

# References


[1] Taoufik Bouezmarni and Jeroen VK Rombouts. Nonparametric density estimation for multivariate bounded data. *Journal of Statistical Planning and Inference*, 140(1):139–152, 2010.

[2] Adrian W. Bowman and Adelchi Azzalini. Applied smoothing techniques for data analysis : the kernel approach with s-plus illustrations. *Journal of the American Statistical Association*, 94:982, 1999.

[3] Paula Branco, Rita P Ribeiro, and Luis Torgo. Ubl: an r package for utility-based learning. *arXiv preprint arXiv:1604.08079*, 2016.

[4] Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM computing surveys (CSUR)*, 49(2):1–50, 2016.

[5] Paula Branco, Luís Torgo, and Rita P Ribeiro. Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pages 36–50. PMLR, 2017.

[6] Paula Branco, Luis Torgo, and Rita P Ribeiro. Pre-processing approaches for imbalanced distributions in regression. *Neurocomputing*, 343:76–99, 2019.

[7] M. Buda, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[8] Luís Camacho, Georgios Douzas, and Fernando Bacao. Geometric smote for regression. *Expert Systems with Applications*, page 116387, 2022.

[9] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[10] Arthur Charpentier and Emmanuel Flachaire. Log-transform kernel density estimation of income distribution. *L'Actualité économique*, 91(1):141–159, 2015.

[11] Arthur Charpentier and Abder Oulidi. Beta kernel quantile estimators of heavy-tailed loss distributions. *Statistics and computing*, 20(1):35–55, 2010.

[12] Song Xi Chen. Probability density function estimation using gamma kernels. *Annals of the Institute of Statistical Mathematics*, 52:471–480, 2000.

[13] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Classbalanced loss based on effective number of samples. *CVPR*, 2019.

[14] Daniela De Angelis and G Alastair Young. Smoothing the bootstrap. *International Statistical Review/Revue Internationale de Statistique*, pages 45–56, 1992.

[15] Daniele De Martini and Fabio Rapallo. On multivariate smoothed bootstrap consistency. *Journal of statistical planning and inference*, 138(6):1828–1835, 2008.

[16] Yifei Ding, Minping Jia, Jichao Zhuang, and Peng Ding. Deep imbalanced regression using cost-sensitive learning and deep feature transfer for bearing remaining useful life estimation. *Applied Soft Computing*, 127:109271, 2022.

[17] Tarn Duong. *Bandwidth selectors for multivariate kernel density estimation*. University of Western Australia Perth, 2004.

[18] Tarn Duong. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of statistical software*, 21:1–16, 2007.

[19] Tarn Duong and Martin L Hazelton. Cross-validation bandwidth matrices for multivariate kernel density estimation. *Scandinavian Journal of Statistics*, 32(3):485–506, 2005.

[20] M Falk and R-D Reiss. Weak convergence of smoothed and nonsmoothed bootstrap quantile estimates. *The Annals of Probability*, pages 362–371, 1989.

[21] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[22] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905, 2018.

[23] Yu Gong, Greg Mori, and Frederick Tung. Ranksim: Ranking similarity regularization for deep imbalanced regression. *arXiv preprint arXiv:2205.15236*, 2022.

[24] Peter Hall, Thomas J DiCiccio, and Joseph P Romano. On smoothing and the bootstrap. *The Annals of Statistics*, pages 692–704, 1989.

[25] Tristen Hayfield and Jeffrey S Racine. Nonparametric econometrics: The np package. *Journal of statistical software*, 27:1–32, 2008.

[26] C. Huang, Y. Li, C. Change Loy, and X. Tang. Learning deep representation for imbalanced classification. *CVPR*, 2016.

[27] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[28] Erin LeDell and Sebastien Poirier. H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*, July 2020.

[29] Lee and Sauchi. Noisy replication in skewed binary classification. *Computational Statistics and Data Analysis*, 34(2):165–191, 2000.

[30] Menardi and Torelli. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28(1):92–122, 2014.

[31] Rita P Ribeiro and Nuno Moniz. Imbalanced regression and extreme value prediction. *Machine Learning*, 109:1803–1835, 2020.

[32] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[33] Snigdha Sen, Krishna Pratap Singh, and Pavan Chakraborty. Dealing with imbalanced regression problem for large dataset using scalable artificial neural network. *New Astronomy*, 99:101959, 2023.

[34] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[35] BW Silverman and GA Young. The bootstrap: to smooth or not to smooth? *Biometrika*, 74(3):469–479, 1987.

[36] Sobom Matthieu Someé. *Estimations non paramétriques par noyaux associés multivariés et applications*. PhD thesis, Université de Franche-Comté, 2015.

[37] Xin Yue Song, Nam Dao, and Paula Branco. Distsmogn: Distributed smogn for imbalanced regression problems. In *Fourth International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 38–52. PMLR, 2022.

[38] Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. Density-based weighting for imbalanced regression. *Machine Learning*, 110:2187–2211, 2021.

[39] Samuel Stocksieker, Denys Pommeret, and Arthur Charpentier. Data augmentation for imbalanced regression. In *International Conference on Artificial Intelligence and Statistics*, pages 7774–7799. PMLR, 2023.

[40] Luis Torgo and Rita Ribeiro. Utility-based regression. In *PKDD*, volume 7, pages 597–604. Springer, 2007.

[41] Luís Torgo, Rita P Ribeiro, Bernhard Pfahringer, and Paula Branco. Smote for regression. In *Portuguese conference on artificial intelligence*, pages 378–389. Springer, 2013.

[42] C. F. J. Wu. Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis. *The Annals of Statistics*, 14(4):1261 – 1295, 1986.

[43] Wenglei Wu, Nicholas Kunz, and Paula Branco. Imbalancedlearningregression-a python package to tackle the imbalanced regression problem. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part VI*, pages 645–648. Springer, 2023.

[44] Y. Yang and Z. Xu. Rethinking the value of labels for improving class-imbalanced learning. *NeurIPS*, 2020.

[45] Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International Conference on Machine Learning*, pages 11842–11851. PMLR, 2021.