Highly Imbalanced Regression with Tabular Data in SEP and Other Applications

Josias K. Moukpe*, Philip K. Chan*, Ming Zhang†
*Department of Electrical Engineering and Computer Science
†Department of Aerospace, Physics and Space Sciences
Florida Institute of Technology, Melbourne, FL, USA
jmoukpe2016@my.fit.edu, {pkc, mzhang}@fit.edu

***Abstract*—We investigate imbalanced regression with tabular data that have an imbalance ratio larger than 1,000 ("highly imbalanced"). Accurately estimating the target values of rare instances is important in applications such as forecasting the intensity of rare harmful Solar Energetic Particle (SEP) events. For regression, the MSE loss does not consider the correlation between predicted and actual values. Typical inverse importance functions allow only convex functions. Uniform sampling might yield mini-batches that do not have rare instances. We propose CISIR that incorporates correlation, Monotonically Decreasing Involution (MDI) importance, and stratified sampling. Based on five datasets, our experimental results indicate that CISIR can achieve lower error and higher correlation than some recent methods. Also, adding our correlation component to other recent methods can improve their performance. Lastly, MDI importance can outperform other importance functions. Our code can be found in https://github.com/Machine-Earning/CISIR.**

***Index Terms*—regression, tabular, highly imbalanced, SEP**

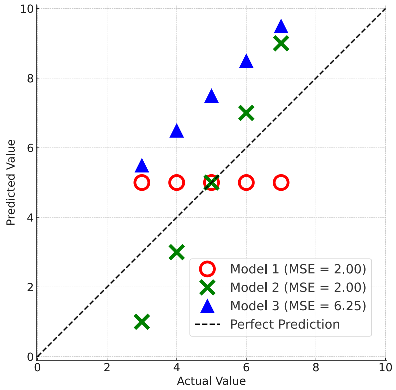


Fig. 1: An issue with MSE.

I. Introduction

This paper tackles highly imbalanced regression for tabular data, a critical and less-explored area compared to image-based regression. Accurately predicting rare-but-important instances is vital in applications like forecasting Solar Energetic Particle (SEP) events to protect astronauts and equipment. We focus on computationally efficient methods that do not rely on synthetic data generation or expensive pretraining.

We identify and address three key challenges. First, the standard Mean Squared Error (MSE) loss is insufficient because it can ignore correlation. As illustrated in Fig. 1, Model 1 (uncorrelated predictions) and Model 2 (positively correlated predictions) both have an identical MSE of 2.0, making them indistinguishable from an error-only perspective. At the same time, correlation alone is also not enough; Model 3 has perfect positive correlation but a much higher MSE. This demonstrates the need to evaluate models on both error and correlation. To achieve this, we propose a weighted Pearson Correlation Coefficient (wPCC) as a supplementary loss term. Second, while re-weighting instances is a common strategy, existing "importance functions" are often rigid (e.g., fixed inverse or linear functions). We introduce Monotonically Decreasing Involution (MDI) importance, a flexible family of functions that can be convex, linear, or concave to suit different datasets. Third, random mini-batch sampling in SGD can exclude rare instances from gradient updates. We employ stratified sampling to ensure rare instances are represented in every batch.

Our integrated approach, CISIR (Correlation, Involution importance, and Stratified sampling for Imbalanced Regression), combines these three solutions. We show that CISIR can achieve lower error and higher correlation than other recent methods, and that its individual components, wPCC and MDI, can effectively improve other approaches.

II. Related Work

Recent methods for imbalanced regression can be clustered into four groups: distribution resampling, label-space smoothing, representation-space calibration, and loss re-weighting. Distribution re-sampling methods directly modify the training data distribution by generating synthetic samples for rare label ranges. SMOGN [1] and SMOTEBoost-R [10] adapt popular synthetic oversampling techniques originally developed for classification tasks to continuous targets.

Label-space smoothing approaches adjust the label distribution or its granularity to alleviate the imbalance. LDS [19] smooths the empirical label density via Gaussian kernel convolution, enabling re-weighting schemes based on a more robust label distribution. HCA [17] constructs a hierarchy of discretized labels at varying granularities, using predictions from coarser levels to refine fine-grained predictions, balancing quantization errors and prediction accuracy.

Representation-space calibration tackles imbalance by enforcing structural regularities directly in latent feature space. FDS [19] aligns latent representations with smoothed label distributions. RankSim [4] explicitly calibrates the representation space to reflect the pairwise ranking structure in labels. ConR [7] introduces a contrastive regularizer that penalizes incorrect proximities in feature space based on label similarity and density, ensuring minority samples remain distinguishable. (For general regression, RnC [20] learns continuous, ranking-aware embeddings by contrasting samples based on their relative ordering in label space. Ordinal Entropy [21] enforces local ordinal relationships via an entropy-based regularize.)

Loss re-weighting (importance) methods rebalance the importance of each instance during training by adapting the regression objective. Inverse-frequency weighting is ubiquitous [19]; DenseLoss [15] employs label-density weighting calibrated linearly with a tunable parameter α. Balanced MSE [13] derives a closed-form objective assuming uniform label distributions, approximating the resulting integral numerically to address imbalance directly within the regression loss.

For imbalanced (long-tailed) classification, various approaches have been proposed, including resampling [23], logit adjustments [9], decoupled training [6], representation learning [5], uniform class separation [8], multiple branches [23], and multiple experts [22].

III. Approach

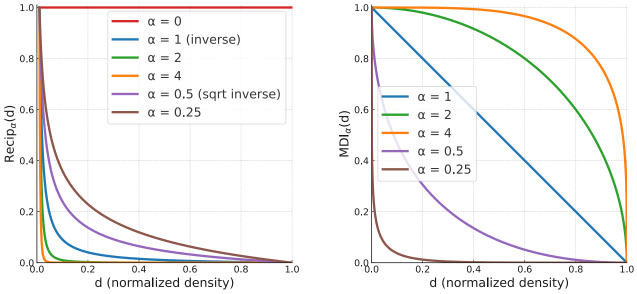
Preliminaries. Let $\{(\mathrm{x}_i, y_i)\}_{i=1}^{N}$ be a training data set $\mathcal{D}$, where $\mathrm{x}_i \in \mathbb{R}^d$ denotes the input of dimensionality d and $y_i \in \mathbb{R}$ is the label, which is a continuous target. We denote $\mathrm{z} = g(\mathrm{x}; \theta_g)$ as the latent representation for x, where $g(\mathrm{x}; \theta_g)$ is the encoder parameterized by a neural network model with parameters θ_g. The final prediction $\hat{y}$ is given by a regressor $f(\mathrm{z}; \theta_f)$ that operates over z, where $f(\cdot; \theta_f)$ is parameterized by θ_f. Therefore, the prediction can be expressed as $\hat{y} = f(g(\mathrm{x}; \theta_g); \theta_f)$.

Highly imbalanced distributions. We divide the targets into equal-width bins. We define the imbalanced frequency ratio as $\rho = freq_{max}/freq_{min}$, where $freq_{max}$ ($freq_{min}$) is the frequency of instances in the largest (smallest non-empty) bin. A distribution is *highly imbalanced* if $\rho \geq 1000$.

Estimating probability density distributions. We use Kernel Density Estimation (KDE) [14] with a Gaussian kernel. The estimated density $\widehat{p}_Y(y)$ for target value y is: $\widehat{p}_Y(y) = \frac{1}{Nh}\sum_j^N K((y - y_j)/h)$, where K is the kernel and h is the bandwidth. KDE is used in DenseLoss [15] and LDS [19].

We denote $\hat{d}_i = \widehat{p}_Y(y_i)$. We normalize all densities into $(0, 1)$: $d_i = \hat{d}_i/(\hat{d}_{\max} + \epsilon)$, where $\hat{d}_{\max}$ is the largest density and ϵ is a small constant (10^{-3}) so that $d_i \neq 1$ (and $MDI(d_i) \neq 0$ in Sec. III-A2). Henceforth d_i refers to the normalized density. We define the imbalance density ratio as $\rho_d = d_{\max}/d_{\min}$, where $d_{\max} = \max_i d_i$ ($d_{\min} = \min_i d_i$,) is the maximum (minimum) normalized density. For KDE, bandwidth h is chosen such that the imbalance density ratio is close to the imbalance frequency ratio; that is, $\rho_d \approx \rho$.

To handle high imbalance, based on KDE, we use the MDI importance function (Sec. III-A) to shift importance away from frequent instances and toward rare instances. To consider correlation in addition to error, we use weighted Pearson Correlation Coefficent (wPCC, Sec. III-B) as a loss regularizer. Stratified Sampling (Sec. III-C) helps produce mini-batches that consistently contain rare samples. Our method is called CISIR, which uses wPCC with MDI in the loss function and stratified sampling for mini-batches (Algorithm 1).



(a) *Recip*   (b) MDI

Fig. 2: *Recip* and MDI importance functions. For *Recip*, we rescale $r_i = Recip_\alpha(d_i)$ so that $r_i \in (0, 1]$ to match MDI.

A. Importance Functions

To encourage that both frequent and rare regions of the feature space are learned well, we attribute importance r_i to instance x_i with normalized density $d_i \in (0, 1)$ based on an *importance* (Imp) function:

$$r_i \;=\; Imp(d_i). \tag{1}$$

We use "importance" to distinguish it from a "weight" in the model. The importance function Imp is a monotonically decreasing function: the lower the normalized density d_i, the higher the resulting importance r_i. The importance can be precomputed once for a dataset and reused during training.

1) Reciprocal Importance: The inverse function [19] is a typical importance function due to its property of balancing the data distribution. To reduce the initial sharp decrease in importance at low density, square-root inverse [19] was also proposed. However, the desirable rate of initial decrease is generally not known and dependent on the dataset, imbalance ratio, and loss function.

To overcome these limitations, we generalize inverse and square-root inverse to a **Reciprocal importance** function that provides more flexibility in the rate of initial decrease in importance at low density. Given the normalized density d_i for instance x_i, we define the reciprocal importance (*Recip*) as:

$$Recip_\alpha(d_i) = \frac{1}{d_i^\alpha}, \quad \alpha \geq 0, \tag{2}$$

where α controls the curvature of the function:

- When $\alpha = 0$, the function reduces to a constant importance of 1 for all instances; ie, no adjustment for imbalance.
- When $\alpha = 1$, it is the typical inverse function $1/d_i$ that achieves the balanced distribution.

- When $\alpha > 1$, it further emphasizes the rare samples beyond the balanced distribution (which is beneficial to some datasets).

Figure 2a shows Reciprocal importance with various α values.

2) Monotonically Decreasing Involution (MDI) Importance: Although Reciprocal importance provides an effective adjustment of importance among frequent and rare instances, it is fundamentally limited by its inherent convexity and exponential form, restricting its ability to represent linear or concave importance relationships. Consequently, Reciprocal importance is insufficient when a more diverse range of importance shapes is needed in various imbalance scenarios.

To address these limitations, we introduce **Monotonically Decreasing Involution (MDI) importance**, a parameterized function with three properties: (1) it is monotonically decreasing, (2) it allows convex, linear, and concave functions to represent a diverse family of importance functions, and (3) it is an involution (or a self-inverse function, where $f(f(x)) = x$) to preserve the same property exhibited by the inverse function that yields a balanced distribution. Given a normalized density d_i for instance x_i, MDI importance is defined as:

$$MDI_\alpha(d_i) = (1 - d_i^\alpha)^{\frac{1}{\alpha}}, \quad \alpha > 0, \tag{3}$$

where α controls the curvature and shape of the function:

- When $0 < \alpha < 1$, the function is convex, similar to Reciprocal importance (Sec. III-A1).
- When $\alpha = 1$, it is linear, similar to DenseLoss [15].
- When $\alpha > 1$, it is concave (which is beneficial to some datasets).
- When $\alpha \gg 1$, it is approximately 1 at low density.

Fig. 2b illustrates MDI importance with various α values.

B. Weighted Pearson Correlation Coefficient (wPCC) as Loss Regularization

For regression, as discussed in Sec. I, MSE does not differentiate models that are equally inaccurate but differ in correlation between the predicted and actual values. Also, while perfect MSE yields perfect positive correlation, perfect positive correlation does not guarantee perfect MSE. Hence, for imbalanced regression, we propose $wMSE$ (weighted MSE) as the primary loss function and $wPCC$ (weighted Pearson Correlation Coefficient) as the secondary loss function or regularization. Each instance i is weighted by importance r_i (Eq. 1), which has been normalized such that $\sum_{i=1}^{N} r_i = 1$. To allow different importance values (e.g. different α values in MDI for $wMSE$ and $wPCC$) for the same instance in the two loss functions, we denote re_i and rc_i as the importance for instance i in $wMSE$ and $wPCC$ respectively. Moreover, re_i and rc_i are obtained from an importance function (Recip or MDI in Sec. III-A) with α values that we denote as α_e and α_c respectively.

We define $wMSE$ as:

$$wMSE \;=\; \sum_{i=1}^{N} re_i \, (y_i - \hat{y}_i)^2, \tag{4}$$

and $wPCC$ as:

$$wPCC \;=\; 1 - \frac{\displaystyle\sum_{i=1}^{N} rc_i \, (y_i - \bar{y}) \, (\hat{y}_i - \bar{\hat{y}})}{\sqrt{\displaystyle\sum_{i=1}^{N} rc_i \, (y_i - \bar{y})^2} \, \sqrt{\displaystyle\sum_{i=1}^{N} rc_i \, (\hat{y}_i - \bar{\hat{y}})^2}}, \tag{5}$$

where $\bar{y}$ and $\bar{\hat{y}}$ are the averages of y and $\hat{y}$ respectively. Our proposed overall loss function is:

$$\mathcal{L} \;=\; wMSE \;+\; \lambda \cdot wPCC, \tag{6}$$

where $\lambda > 0$ adjusts the influence of $wPCC$.

Moreover, $wPCC$ can help reduce $wMSE$. Following the MSE decomposition in [11, Eq. 9], we have:

$$\begin{aligned} \mathrm{MSE}(\hat{y}, y) = \big(\bar{y} - \bar{\hat{y}}\big)^2 + \mathrm{var}(\hat{y}) + \mathrm{var}(y) \\ - 2 \cdot \mathrm{cov}(\hat{y}, y), \end{aligned} \tag{7}$$

which, after some transformation, yields:

$$\begin{aligned} \mathrm{MSE}(\hat{y}, y) = \big(\bar{y} - \bar{\hat{y}}\big)^2 + \big(\mathrm{sd}(\hat{y}) - \mathrm{sd}(y)\big)^2 \\ + 2 \; \mathrm{sd}(\hat{y}) \; \mathrm{sd}(y) \, \Big(1 - \mathrm{PCC}(\hat{y}, y)\Big). \end{aligned} \tag{8}$$

where $\mathrm{var}(\cdot)$, $\mathrm{cov}(\cdot, \cdot)$, and $\mathrm{sd}(\cdot)$ denote the variance, covariance, and standard deviation operators, respectively.

Eq. (8) is a sum of three terms: a term for mismatch in the mean, $(\bar{y} - \bar{\hat{y}})^2$; a second term for mismatch in the standard deviation, $(\mathrm{sd}(\hat{y}) - \mathrm{sd}(y))^2$; and a third term, $2 \; \mathrm{sd}(\hat{y}) \; \mathrm{sd}(y) \, (1 - \mathrm{PCC}(\hat{y}, y))$, related to the correlation deficit. While the first two terms encourage matching the moments of the data distribution, the third term's minimization highlights a critical point. MSE can be reduced by increasing the Pearson Correlation Coefficient, $\mathrm{PCC}(\hat{y}, y)$, towards 1, and/or by decreasing the standard deviation of the predictions, $\mathrm{sd}(\hat{y})$ towards 0.

In general, since the second term is present and $\mathrm{sd}(y) \neq 0$, the prediction standard deviation $\mathrm{sd}(\hat{y})$ does not collapse to zero. However, in some cases, illustrated by Model 1 (with no correlation) and Model 2 (with some correlation) in Fig. 1, MSE cannot distinguish the two models. This is caused by $\mathrm{sd}(\hat{y})$ of Model 1 being zero, which renders the third term to be zero and the lack of correlation to be ignored. Similarly, a small $\mathrm{sd}(\hat{y})$ diminishes the penalty due to poor correlation. To address this limitation, our proposed $wPCC$ regularizer directly penalizes poor correlation by minimizing $1 - \mathrm{PCC}(\hat{y}, y)$.

C. Stratified Sampling in Mini-Batches (SSB)

Training neural networks with stochastic gradient descent (SGD) typically involves partitioning the dataset $\mathcal{D}$ into mini-batches of size B, yielding $M = N/B$ mini-batches and parameter updates per epoch. With highly imbalanced data, uniformly sampled mini-batches may not represent rare target regions adequately. Consider the probability of rare samples as $\pi_r \approx 0$, the probability of no rare instances in a uniformly drawn mini-batch is $(1 - \pi_r)^B$, which is close to 1 when B is relatively small compared to $1/\pi_r$. That is, some mini-batches

Algorithm 1 CISIR Training Procedure

```
Require: Dataset $\mathcal{D}$, batch size $B$, hyperparameters $\alpha_e, \alpha_c, \lambda$,
    learning-rate $\eta$
 1: Estimate densities $d_i$ with KDE (Sec. III)
 2: Compute importance $re_i, rc_i$ using MDI (Eq. (3))
 3: Form $B$ groups with Stratified Sampling (Sec. III-C)
 4: while model not converged do
 5:     Build mini-batch $\mathcal{B}$ by drawing one sample from each
        group
 6:     Forward-propagate to obtain predictions $\hat{y}_i$
 7:     Compute $wMSE$ (Eq. (4)) and $wPCC$ (Eq. (5))
 8:     $\mathcal{L} \leftarrow wMSE + \lambda \cdot wPCC$          (Eq. (6))
 9:     Back-propagate $\nabla_\theta \mathcal{L}$ and update model parameters
        with step $\eta$
10: end while
11: return Trained model parameters $\theta$
```

might not have rare instances at all, which lead to gradients that do not reduce loss for rare instances during some model updates.

To ensure rare instances are represented across mini-batches, we propose to perform stratified sampling to form mini-batches such that each mini-batch has a similar distribution as the overall training distribution. Consequently, gradients computed from these stratified mini-batches approximate the gradient computed over the training set, and every model update reduces loss for some rare instances.

To perform stratified sampling, we first choose M such that M is less than or equal to the number of rare instances. We sort all instances based on their target values. The sorted instances are then divided into B groups, each containing M instances (The final group may have fewer than M instances). The number of groups B matches the size of a mini-batch, with each group contributing one instance to the mini-batch. To create each mini-batch, we randomly select one instance from each of the B groups. For example, if larger target values are rarer, the M instances with the largest target values are in one group. Each of the M instances is randomly assigned to one of M mini-batches.

Algorithm 1 summarizes the overall CISIR method. We estimate densities via KDE on line 1. We then compute the importance values using MDI on line 2. With stratified sampling, we form B groups of data instances on line 3. From lines 4 to 9, we iteratively train the model by getting a mini-batch on line 5, taking the model's predictions on line 6, computing the loss with lines 7-8, and updating the model parameters on line 9. After convergence, the trained model is finally returned on line 11.

IV. Experimental Evaluation

A. Datasets and Evaluation Metrics

We evaluate our method on five highly imbalanced datasets: our SEP-EC (proton intensity change) and SEP-C (peak proton intensity) datasets, and three public benchmarks: SARCOS (robot arm torque) [16], Blog Feedback (BF, comment forecasting) [2], and Online News Popularity (ONP, article shares) [3]. Detailed statistics for each, including the imbalance ratio (ρ), are in Table III. Our SEP datasets are available at https://huggingface.co/datasets/Machine-Earning/CISIR-datasets/resolve/main/CISIR-data.zip.

Our primary evaluation metrics are two hybrid scores that balance overall and rare-instance performance: $AORE = (MAE + MAE_R)/2$ and $AORC = (PCC + PCC_R)/2$. Here, MAE and PCC are the overall Mean Absolute Error and Pearson Correlation Coefficient, while MAE_R and PCC_R are calculated on a subset of important rare instances. We prioritize minimizing $AORE$ over maximizing $AORC$. For the SEP-EC dataset, this rare subset specifically includes only instances with positive (increasing) intensity change. We use standard error to assess statistical significance.

B. Baseline Methods and Experimental Procedures

Experimental Setup. Our primary baselines are DenseLoss [15], Balanced MSE [13], and the effective SQINV+LDS+FDS variant [19], chosen for their computational efficiency. We also evaluate a CISIR variant, Recip+wPCC+SSB, to compare importance functions. We use the official implementations and hyperparameters for LDS/FDS [18] and Balanced MSE [12] and implement DenseLoss ourselves. Our source code is available at https://github.com/Machine-Earning/CISIR. All models are implemented in TensorFlow using a residual MLP architecture. Dataset-specific hyperparameters are determined via four-fold stratified cross-validation on the training data. If official splits are unavailable, we create a stratified 2/3–1/3 train-test split. Models are trained to convergence with the Adam optimizer, early stopping, and a learning rate scheduler. All reported results are averages over five runs with fixed random seeds for reproducibility.

C. Results

Our main results in Table I show that CISIR generally outperforms recent methods. It achieves the best AORE on four of the five datasets and top-two AORC scores across all of them. This strong performance extends to important rare instances, where CISIR and its variants also demonstrate improved error (MAE_R) and correlation (PCC_R).

Analysis of CISIR's core components confirms their effectiveness. First, incorporating our $wPCC$ regularizer into competing methods consistently improves both their AORE and AORC across nearly all settings (Table II), demonstrating its general utility for both error reduction and correlation enhancement. Second, a direct comparison of importance functions reveals that our proposed MDI and Recip functions outperform standard inverse-weighting schemes (Table III). The flexibility of MDI to use concave functions ($\alpha_e > 1$) was beneficial for the SEP-C and ONP datasets, while Recip's ability to further emphasize rare instances ($\alpha_e > 1$) was key for SARCOS and SEP-EC. We find no clear relationship between the optimal α_e and a dataset's imbalance ratio ρ, highlighting the need for tunable importance functions rather than fixed ones. The weaker performance of

TABLE I: CISIR vs. recent methods (**bold** = best, underline = 2nd best, * = statistically significant)

Dataset	Method	MAE↓	MAE_R ↓	AORE↓	PCC↑	PCC_R ↑	AORC↑
SEP-EC	SQINV+LDS+FDS	0.177	0.566*	0.371	-0.025	0.067	0.021
	BalancedMSE	0.161	0.659	0.410	-0.036	0.141	0.053
	DenseLoss	**0.071***	0.626	0.348*	**0.286**	0.699*	**0.493**
	Recip+wPCC+SSB	0.089*	0.606	0.348*	0.219	0.699*	0.459
	CISIR	0.184	**0.441***	**0.313***	0.274*	**0.703**	0.488*
SEP-C	SQINV+LDS+FDS	1.681	4.314	2.997	0.173	0.481	0.327
	BalancedMSE	1.683	3.614	2.649	0.394	0.393	0.393
	DenseLoss	**0.237***	2.245	1.241	0.690*	0.588	0.639
	Recip+wPCC+SSB	1.173	**1.376***	1.274	0.627	**0.661***	0.644
	CISIR	0.335*	1.875*	**1.105***	**0.702**	0.593	**0.647**
SARCOS	SQINV+LDS+FDS	0.575	0.748	0.661	0.020	-0.049	-0.015
	BalancedMSE	0.571	1.694	1.132	0.189	-0.170	0.010
	DenseLoss	0.058	0.076	0.067	0.964	0.830	0.897
	Recip+wPCC+SSB	**0.053**	0.071	**0.062**	**0.986**	**0.910***	**0.948***
	CISIR	0.055	**0.069**	**0.062**	0.982*	0.876*	0.929*
BF	SQINV+LDS+FDS	1.036	1.780	1.408	-0.152	0.065	-0.044
	BalancedMSE	0.689	1.769	1.229	-0.011	0.066	0.028
	DenseLoss	**0.169***	0.747	**0.458**	0.735	0.301	0.518
	Recip+wPCC+SSB	0.187*	0.740	0.463*	0.733	0.319*	0.526
	CISIR	0.280	**0.709***	0.495	**0.737**	**0.330***	**0.533**
ONP	SQINV+LDS+FDS	2.628	4.438	3.533	0.034	-0.012	0.011
	BalancedMSE	2.798	4.154	3.476	-0.028	0.012	0.047
	DenseLoss	**0.317**	1.311	0.814	**0.325***	0.054	0.189
	Recip+wPCC+SSB	0.326	1.351	0.838	0.288	**0.095**	0.192
	CISIR	0.379	**1.180***	**0.780**	0.299*	0.093*	**0.196**

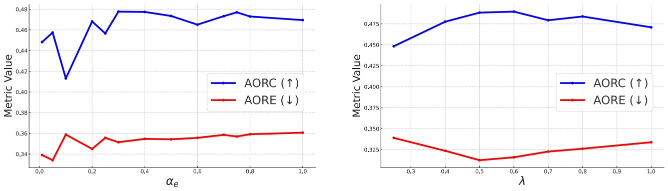
TABLE II: Incorporating wPCC into other methods (**bold** = best, * = statistically significant)

Method	SEP-EC		SEP-C		SARCOS		BF		ONP	
	AORE↓	AORC↑	AORE↓	AORC↑	AORE↓	AORC↑	AORE↓	AORC↑	AORE↓	AORC↑
SQINV+LDS+FDS	**0.371**	0.021	2.997	0.327	0.661	−0.015	1.408	−0.044	3.533	0.011
+ wPCC	**0.371**	**0.270***	**2.891**	**0.411***	**0.657**	**0.051***	**1.062***	**0.130***	**3.416***	**0.038**
BalancedMSE	0.410	0.053	2.649	0.393	1.132	0.010	1.229	0.028	3.476	0.047
+ wPCC	**0.392**	**0.057**	**2.484**	**0.491***	**0.884***	**0.097***	**0.990**	**0.162***	**3.223***	**0.086***
DenseLoss	0.348	0.493	1.241	0.639	0.067	0.897	**0.458**	0.518	0.814	0.189
+ wPCC	**0.344**	**0.500**	**1.084**	**0.660**	**0.062**	**0.920***	**0.458**	**0.529**	**0.791**	**0.203**

baselines like SQINV+LDS+FDS may be due to the absence of a correlation-based regularizer like $wPCC$ and the use of a rigid importance scheme (e.g., SQINV is equivalent to Recip with a fixed $\alpha = 0.5$).

D. Analyses

Ablation study. Results from an ablation study on CISIR with the SEP-EC dataset are in Table IV. We observe that each of the 3 proposed components (MDI, wPCC, and SSB) contributes to CISIR. Particularly, wPCC contributes the most. Moreover, wPCC not only increases correlation (AORC), it also reduces error (AORE). This indicates that the constraint from wPCC in the loss helps achieve a lower local minimum for wMSE during training.



(a) Sensitivity to α_e (b) Sensitivity to λ

Fig. 3: Sensitivity analysis of CISIR to α_e and λ on SEP-EC.

Parameter Sensitivity. We analyze the sensitivity of CISIR's key hyperparameters on the SEP-EC dataset. The parameter α_e, which controls instance importance for wMSE, reveals

TABLE III: Comparison of importance functions (**bold** = best, underline = 2nd best, * = statistically significant)

Importance functions	AORE↓				
	SEP-C	SARCOS	ONP	SEP-EC	BF
Imb. ratio ρ	*1,476*	*3,267*	*6,746*	*10,478*	*33,559*
INV	2.456	0.064	0.849	0.362	0.451
SQINV	1.313	0.068	0.765	0.369	0.447
DenseLoss	1.277	0.065	0.842	0.355	0.471
Recip	1.289	0.063	**0.764**	0.349	**0.445**
MDI	**1.105***	**0.062**	0.780	**0.313***	0.495
α_e in Recip	0.70	1.10	0.90	1.20	0.70
α_e in MDI	2.40	0.20	1.10	0.01	0.50

TABLE IV: Ablation study of CISIR on the SEP-EC dataset (**bold** = best, underline = 2nd best)

Method	AORE↓	AORC↑
CISIR	0.313	0.488
w/o MDI (with INV)	0.362 (+0.049)	0.447 (-0.041)
w/o wPCC	1.160 (**+0.847**)	0.194 (**-0.294**)
w/o SSB	0.333 (+0.020)	0.463 (-0.025)

a trade-off: as α_e increases, AORE degrades while AORC improves (Fig. 3a). We thus select $\alpha_e < 0.1$ to prioritize error reduction by up-weighting rare instances. For α_c, which governs importance for wPCC, we find that uniform importance ($rc_i = 1$) is effective for most datasets. SEP-C is a notable exception, benefiting from an atypical concave importance function with $\alpha_c = 1.7$. In contrast to the trade-off with α_e, the wPCC contribution weight, λ, allows for co-optimization. Both AORE and AORC are jointly optimized when $\lambda \in [0.5, 0.6]$ (Fig. 3b), confirming that our wPCC regularizer simultaneously reduces error and increases correlation.

More analyses are included in the longer version of this paper on arxiv.org

V. Conclusion

For highly imbalanced regression with tabular data, we propose CISIR that incorporates wPCC as a secondary loss function, MDI importance that allows convex, linear, and concave functions, and stratified sampling in the mini-batches. Our experimental results indicate that CISIR can achieve lower error and higher correlation than some recent methods. Also, adding our wPCC component to other methods is beneficial in not only improving correlation, but also reducing error. Lastly, MDI importance can outperform other importance functions.

References

[1] 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*, 2017.

[2] Krisztian Buza. Feedback prediction for blogs. In *Data Analysis, Machine Learning and Knowledge Discovery*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 145–152. Springer, Cham, 2014.

[3] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In *Proceedings of the 17th Portuguese Conference on Artificial Intelligence (EPIA 2015), Progress in Artificial Intelligence*, volume 9273 of *Lecture Notes in Computer Science*, pages 535–546. Springer, 2015.

[4] Yu Gong, Greg Mori, and Fred Tung. Ranksim: Ranking similarity regularization for deep imbalanced regression. In *International Conference on Machine Learning*, 2022.

[5] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2021.

[6] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.

[7] Mahsa Keramati, Lili Meng, and R David Evans. Conr: Contrastive regularizer for deep imbalanced regression. In *International Conference on Learning Representations*, 2024.

[8] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogerio S Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6918–6928, 2022.

[9] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

[10] Nuno Moniz, Rita Ribeiro, Vitor Cerqueira, and Nitesh Chawla. Smoteboost for regression: Improving the prediction of extreme values. In *2018 IEEE 5th international conference on data science and advanced analytics (DSAA)*, 2018.

[11] Allan H Murphy. Skill scores based on the mean square error and their relationships to the correlation coefficient. *Monthly weather review*, 116(12):2417–2424, 1988.

[12] Jiawei Ren. Balanced mse for imbalanced regression, 2022. GitHub repository: https://github.com/jiawei-ren/BalancedMSE.

[13] Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. Balanced mse for imbalanced visual regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[14] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[15] Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. Density-based weighting for imbalanced regression. *Machine Learning Journal*, 2021.

[16] Sethu Vijayakumar and Stefan Schaal. SARCOS robot-arm inverse dynamics dataset. Supplementary material to "LWPR: An O(n) Algorithm for Incremental Real-Time Learning in High-Dimensional Space", 2000.

[17] Haipeng Xiong and Angela Yao. Deep imbalanced regression via hierarchical classification adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[18] Yuzhe Yang. Delving deeper in imbalanced regression (lds/fds), 2021. GitHub repository: https://github.com/YyzHarry/imbalanced-regression.

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

[20] Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: Learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 2023.

[21] Shihao Zhang, Linlin Yang, Michael Bi Mi, Xiaoxu Zheng, and Angela Yao. Improving deep regression with ordinal entropy. In *The Eleventh International Conference on Learning Representations*, 2022.

[22] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. *Advances in neural information processing systems*, 35:34077–34090, 2022.

[23] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.