SLACE: A Monotone and Balance-Sensitive Loss Function for Ordinal Regression

Inbar Nachmani[1], **Bar Genossar**[1], **Coral Scharf**[1], **Roee Shraga**[2], **Avigdor Gal**[1]

[1]Technion – Israel Institute of Technology
[2]Worcester Polytechnic Institute
{inbar.nac, sbargen, coralscharf}@campus.technion.ac.il, rshraga@wpi.edu, avigal@technion.ac.il

Abstract

Ordinal regression classifies an object to a class out of a given set of possible classes, where labels possess a natural order. It is relevant to a wide array of domains including risk assessment, sentiment analysis, image ranking, and recommender systems. Like common classification, the primary goal of ordinal regression is accuracy. Yet, in this context, the severity of prediction errors varies, e.g., in risk assessment, Critical Risk is more urgent than High risk and significantly more urgent than No risk. This leads to a modified objective of ensuring that the model's output is as close as possible to the correct class, considering the order of labels. Therefore, ordinal regression models should use ordinality-aware loss for training. In this work, we focus on two properties of ordinality-aware losses, namely monotonicity and balance sensitivity. We show that existing ordinal loss functions lack these properties and introduce SLACE (Soft Labels Accumulating Cross Entropy), a novel loss function that provably satisfies said properties. We demonstrate empirically that SLACE outperforms the state-of-the-art ordinal loss functions on most tabular ordinal regression benchmarks.

Code, Datasets, Extended version —
https://github.com/inbarnachmani/SLACE

Introduction

Ordinal regression (also known as ordinal classification (Gutiérrez et al. 2015)) classifies an object to a class out of a given set of possible classes, where labels possess a natural order. It is of interest in the research community at large, with applications in risk assessment (Alicioglu, Sun, and Ho 2020), sentiment analysis (Saad and Yang 2019), image ranking (Liu et al. 2011), and closely related fields such as recommender systems (Koren and Sill 2011).

Ordinal regression can be positioned as a classification problem, yet it differs from multi-class classification. In particular, the distance between the model answer and the correct answer matters. It also differs from ranking since we care about concrete class assignments. In addition, while the distance between answers matters, ordinal regression differs from value prediction (regression) since the class is taken from a fixed and discrete set of classes. Specifically, distance by itself may not suffice, and the severity of diverging from the correct answer may depend on where divergence occurs on the ordinal scale.

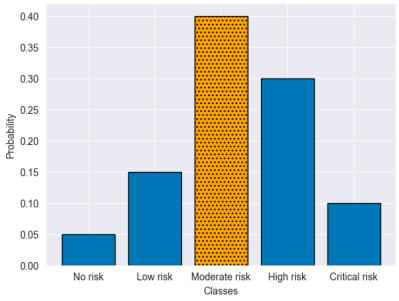


Figure 1: Child maltreatment classification system example.

As an illustrative example, consider a child maltreatment classification system, assessing children's risk at home on a five-scale classification of No risk (1), Low risk (2), Moderate risk (3), High risk (4) and Critical risk (5). For this setup, it is clear that a Critical risk is more urgent than High risk and significantly more urgent than No risk and associated classes confirm that. Figure 1 shows a possible classifier output as softmax probability, where Moderate risk is the true class. While classifiers mostly care about the argmax result, ordinal regression is also affected by the distribution around the chosen class. In this case, classes 2 and 4, which are closer to class 3, have a higher probability than 1 and 5.

In recent years, numerous works dealt with ordinal regression (Diaz and Marathe 2019; Castagnos, Mihelich, and Dognin 2022; Albuquerque, Cruz, and Cardoso 2021; Kasa et al. 2024), typically assuming a constant distance between classes, which may not accurately reflect real-world scenarios. For instance, in the child mistreatment example, Moderate risk may be closer to Low risk in terms of informational content compared to No risk is to Low risk. Consequently, misclassifying an assessment as Moderate risk rather than Low risk may be less acute than misclassifying it as No risk, which may result in dropping a child's case altogether. Referring to Figure 1, since Moderate risk is the true class, a distribution with higher probability assigned to the High Risk class compared to the Low Risk class is preferable since High Risk is closer to Moderate

Risk than Low Risk. In this work, we define two desirable properties of an ordinal regression loss function, namely monotonicity and balance-sensitivity. We analyze existing loss functions, showing their inability to support such properties and present SLACE (**S**oft **L**abel **A**ccumulating **C**ross **E**ntropy), a novel ordinal loss function that provably support both monotonicity and balance-sensitivity. Using a thorough empirical analysis, we show that SLACE outperforms state-of-the-art ordinal loss functions on most tabular ordinal regression benchmarks. The paper offers the following specific contributions:

- We propose a desirable set of proprieties an ordinal loss function should fulfill.
- We present a novel loss function (SLACE) that can be easily adapted to different tasks and machine learning models and satisfies the desired properties.
- We show empirically, using known tabular benchmarks, SLACE superiority over existing loss functions.

Related Work

Frank and Hall (2001) were the first to address the ordinal regression problem, training a set of independent binary classifiers (one per class) and selecting the most probable class. Rennie and Srebro (2005) cast the problem as a regression problem with discrete ordered labels and modified classification loss functions accordingly. Others, *e.g.*, (Vanbelle and Albert 2009; Baccianella, Esuli, and Sebastiani 2009) extended nominal methods to address the problem.

Several loss-sensitive classification approaches proposed loss function designs that imposes a higher penalty for greater discrepancies between predictions and labels. Key contributions include SORD (Diaz and Marathe 2019), OLL (Castagnos, Mihelich, and Dognin 2022), CO2, and HO2 (Albuquerque, Cruz, and Cardoso 2021). These works typically assume a fixed distance between neighboring classes, failing to adequately capture the relationships between classes in the dataset. We compare state-of-the-art ordinal losses with SLACE, showing the latter is superior in terms of desired properties. We further show SLACE empirical dominance over multiple benchmarks.

Amigo et al. (2020) proposed an effectiveness proximity-aware closeness metric for ordinal classifiers using class proximity (CEM), evaluating existing methods with it. In this work, we propose a novel method to embed proximity-awareness into a loss function, demonstrating its superior performance over state-of-the-art

Ordinal relationships induce different levels of error severity. Researchers of fine-grained classification also looked at varying error severity. Collins, Rozanov, and Zhang (2018) pointed at the challenge in settings where classes share different levels of semantic similarity, making prediction harder. Suresh and Ong (2021) suggest a label-aware contrastive loss, embedding class relationships into their loss function, thus allowing a distinction among different levels of negatives examples (*i.e.*, different levels of error severity). However, this approach overlooks the inherent label ordinality and necessitates training an additional neural network with a contrastive loss.

Similar to distance learning between labels in fine-grained classification, there are also works that focus on learning distances to re-represent ordinal features. Shi, Li, and Sha (2016) suggest an algorithm for re-representation of ordinal features with learned distances. Zhang and Cheung (2020) suggest a clustering algorithm that learns distances for ordinal features. While these methods do acknowledge ordinality, they do not confront the problem of ordinal classification and necessitates training additional algorithms.

Class imbalance is a long standing problem (Wang et al. 2021; Chawla, Japkowicz, and Kotcz 2004). Balance-sensitivity in ordinal data is of recent interest (Domingues et al. 2018; Lázaro and Figueiras-Vidal 2023). To the best of our knowledge, we are the first to embed balance-sensitivity concerns in a loss function for ordinal regression.

Model and Problem Definition

Ordinal regression: In an ordinal regression task, an input vector x is classified into a set of classes $O = \{c_1, \ldots, c_K\}$, with induced total order such that $c_1 < c_2 < \ldots < c_K$, where $K \in \mathbb{N}$ (typically $K > 2$). We denote by $g(x) \in O$ the true label of x.

The output of the classification task can be either a probability distribution over the classes $p(x) \in [0,1]^K$ or a predicted class $s(x) \in O$ (we can get $s(x)$ from $p(x)$ using $s(x) = argmax_{c_i \in O}(p(x)_i)$). An element in the vector, $p(x)_i$ ($i \in \{1, \ldots, K\}$) represents the probability for the i-th class to be the correct one as assigned by the model.

Loss: Given a loss function $loss : p(x) \rightarrow \mathbb{R}$, which quantifies the discrepancy between predicted and true labels, the goal of an ordinal regression task is to train a model that minimizes $loss$ over a set of training items.

A naïve choice of a loss function could be the common multi-class Cross Entropy (CE), defined as follows.

$$CE(x,p) = -\sum_{i=1}^{K} \mathbb{1}(g(x) = c_i) \log(p(x)_i) \qquad (1)$$

where $\mathbb{1}(\cdot)$ denotes an indicator function.

Such a loss function disregards ordinality, contributing the same amount of error to a classification loss regardless of its ordinal distance. Using our running example (Figure 1), CE assigns the same loss when predicting either High risk or No risk when the true class is Moderate risk.

An ordinal extension of CE involves using soft labels (SORD) (Diaz and Marathe 2019; Bertinetto et al. 2019). Soft labels indicate the degree of membership of the training data to classes, rather than using a one-hot encoding.

$$SORD(x,p) = -\sum_{i=1}^{K} q(x)_i \log(p(x)_i) \qquad (2)$$

where

$$q(x)_i = \frac{\exp(-\alpha \cdot dis(c_i, g(x)))}{\sum_{j=1}^{K} \exp(-\alpha \cdot dis(c_j, g(x)))} \qquad (3)$$

and dis is a distance function, typically defined to be $dis(c_i, c_j) = |i - j|$.

Closeness and distance: Given a closeness metric $\mathcal{C}(c_i, c_j)$, a desired outcome favors classes that are closer to the true class. Hence, if $\mathcal{C}(c_i, g(x)) > \mathcal{C}(c_j, g(x))$, predicting c_i is prefered over c_j.

Closeness is largely defined by the absolute distance between classes, where smaller values indicate closer classes, such as in MSE (mean squared error), MAE (mean absolute error), SORD (Eq. 2) and more (Castagnos, Mihelich, and Dognin 2022; Gutiérrez, Pérez-Ortiz, and Suárez 2017). This means that the distance between classes c_i and c_j is given by $dis(c_i, c_j) = |i - j|$. An inferred closeness metric could be defined as $\mathcal{C}(c_i, c_j) = K - |i - j|$, reflecting the intention for classes with greater distances to exhibit smaller closeness values, and vice versa. Unless mentioned otherwise, we consider distance and closeness to be as mentioned when analyzing loss functions.

Desiderata for Ordinal Regression Loss

According to the proper scoring rule (PSR) (Gneiting and Raftery 2007; Merkle and Steyvers 2013), a loss function should be minimized when $p(x)$ is a one hot vector, matching the ground truth. Also, a loss function should be convex for easier optimization (Kawaguchi, Huang, and Kaelbling 2019; Lázaro and Figueiras-Vidal 2023) (Convexity (Cx)). We aim to design a loss function for ordinal regression tasks that, in addition to the PSR and Cx properties, satisfies three principal desiderata, as follows.

- **Residue Minimization:** Ordinal regression aims at minimizing distance between chosen and correct classes.
- **Popularity Variation:** Impact of distance between chosen and correct classes depends on class popularity.
- **Proportional Confidence:** The distribution of confidence in an ordinal classifier's decision making should be minimized proportionally to the proximity of class to the correct class and class popularity.

The presented desiderata specifies, beyond the PSR and Cx, how classification error should behave. The following two properties, namely *monotonicity*, focusing on class ordinality, and *balance-sensitivity*, which tackles class imbalance, offer a concretization of the three principals.

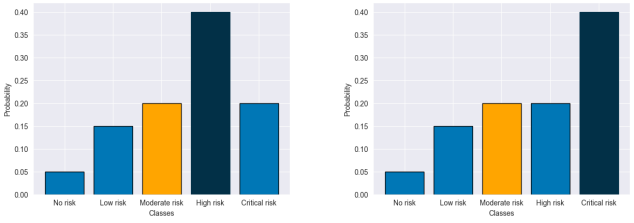
Monotonicity

For ordinal regression, we expect the loss to penalize more misclassifications that are further away from the true class, ensuring that predictions closer to the true class are preferred. To formalize this we propose ordinal monotonicity.

Property 1 (Ordinal Monotonicity)**.** *Let x be an input item and $p(x) = \{p_1, ..., p_K\}$ and $p'(x) = \{p'_1, ..., p'_K\}$ be two probability distributions over predicted class set O s.t. $\exists i \neq j \in \{1, ..., K\}$ for which $\mathcal{C}(c_i, g(x)) > \mathcal{C}(c_j, g(x))$ and $p'_i = p_i + \epsilon, p'_j = p_j - \epsilon$ for $\epsilon \in (0, p_j]$, and $\forall h \neq i, j : p'_h = p_h$. Then, a loss function $Loss$ is* ϵ-ordinal monotonic *if*

$$Loss(x, p') < Loss(x, p). \tag{4}$$

Property 1 states that a loss function is ϵ-ordinal monotonic if changing the probabilities of the predictions, with the probability mass moving closer to the true class, results in a loss decrease. Specifically, we require that for two system outputs, if the difference between prediction probability lies in only two classes, the system with a probability distribution closer to the true class yields a lower loss.



(a) more mass around g(x). (b) less mass around g(x).

Figure 2: (a)'s probability distribution is preferred over (b). Moderate risk is the true class, and s(x) is in dark blue.

To illustrate the importance of monotonicity, consider Figure 2. Let x be an item with $g(x) =$ Moderate risk, and let $p'(x)$ and $p(x)$ be distributions as shown in figures 2a and 2b, respectively, where $p'(x)$ differs from $p(x)$ by a 0.2 mass shift from Critical risk to High risk. A loss function that satisfies Property 1 produces a lower loss value for $p'(x)$ than $p(x)$, given $g(x)$, as the mass shifted from Critical risk to High risk, closer to the true class Moderate risk.

The following proposition ensures that a loss function that satisfies ordinal monotonicity also satisfies PSR.

Proposition 1. *An ϵ-ordinal monotonic loss function $Loss$ satisfies:*

$$argmin_{p(x)}(Loss(x, p)) = e^{g(x)} \tag{5}$$

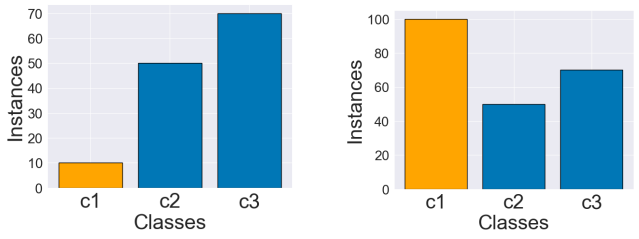
where $e^{g(x)}$ is the one-hot ground truth vector such that, for $g(x) = c_i$, $e_i^{g(x)} = 1$, $\forall j \neq i$, $e_j^{g(x)} = 0$.

Proof. Let $p(x)$ be an output distribution for item x, different from the ground truth vector. Let $p'(x)$ be $p(x)$ after mass shift as defined in Property 1 from any class c_j s.t. $p_j \neq 0$ and $c_j \neq g(x)$ to $g(x)$ where $\epsilon = p_j$. Therefore, $p'_j = 0, p'_i = p_i + p_j$. From Property 1 we get that $Loss(x, p') < Loss(x, p)$. Now, we can recursively define $p(x) = p'(x)$ until $p'_j = 0$ for every $c_j \neq g(x)$, which means that $p'_i = 1$. Therefore, $p'(x)$ is the ground truth vector and comparing to the original output distribution we get that the ground truth vector has a lower loss. □

Balance-sensitivity

Rarer classes inherently carry more information (Amigo et al. 2020). Consequently, distancing predictions from a rarer class leads to a greater loss of information compared to a larger class. To illustrate, consider two cases of true classes, with 10 instances (Figure 3a), and 100 instances (Figure 3b), respectively. A single instance in the first case would be a tenth of the overall information we have on the class while in the second case it is only a hundredth. Consequently, errors regarding an instance in the first case are more damaging. Translated into loss functions, distancing

the model's prediction from a small class should be more detrimental than from a larger class, taking into account ordinality and the probability distribution assigned by a model, rather than just the final prediction.



(a) small g(x) class. (b) big g(x) class.

Figure 3: Label distributions where $g(x) = c_1$ and the only difference is amount of instances in $g(x)$.

Property 2 (balance-sensitivity)**.** *Let D and D^* be class distributions on a set of ordered classes O. There exists an index $i \in O$ where $g(x) = c_i$, such that $|c_i| + m = |c_i^*|$ for some $m > 0$, and for every $j \neq i$ with $j \in \{1, \ldots, K\}$, it holds that $|c_j| = |c_j^*|$.*

Given $g(x)$, $p(x)$, and $p'(x)$ where $g(x) = c_i$:

- *$p(x)_i - \epsilon = p'(x)_i$ for some $\epsilon > 0$,*
- *For $l = arg\max_{h \neq i} \mathcal{C}(c_h, c_i)$, the closest class index to i, $p(x)_l + \epsilon = p'(x)_l$,*
- *For every $j \neq i, l$, $p(x)_j = p'(x)_j$,*

A loss function $Loss$ satisfies the balance-sensitivity *property if*

$$Loss_D(x, p') - Loss_D(x, p) > Loss_{D*}(x, p') - Loss_{D*}(x, p) \quad (6)$$

where $Loss_D$ is the loss given class distribution D.

A loss function $Loss$ satisfies Property 2 if moving model prediction further from a small class will have a bigger effect on the loss than a big class. For example, consider Figure 3. By moving prediction probability from c_1 to c_2 we would loss more information about the class in Figure 3a than in Figure 3b, and therefore the more severe errors in Figure 3a should be reflected by a higher loss.

SLACE

We are now ready to present SLACE, a monotone and balance-sensitive loss function for ordinal regression. We first present a proximity-aware closeness and distance, followed by the accumulating cross-entropy loss function, which jointly satisfy monotonicity and balance-sensitivity.

Proximity-Aware Closeness and Distance Function

Closeness was defined under the assumption of a constant distance between classes, without taking into account the unique properties of the dataset. We propose to integrate a proximity measure (Amigo et al. 2020) into the distance function to tune the loss to the specifics of the dataset. Proximity between two classes is defined as follows.

$$\text{prox}_{c,c'} = \begin{cases} -\log\left(\frac{1}{n}\left(\frac{n_c}{2} + \sum_{j=c+1}^{c'} n_j\right)\right), & \text{if } c < c' \\ -\log\left(\frac{1}{n}\left(\frac{n_c}{2}\right)\right), & \text{if } c = c' \\ -\log\left(\frac{1}{n}\left(\frac{n_c}{2} + \sum_{j=c'}^{c-1} n_j\right)\right), & \text{if } c > c' \end{cases} \quad (7)$$

where n_j is the number of elements in class j, j is the summation running index, and $n = \sum_{j \in O} n_j$. $prox$ captures class informational closeness between classes c and c' as the prior probability of an item to belong to all classes between (and including) them. For an item x, given a prediction $s(x)$ and a true class $g(x)$, the larger $\text{prox}(s(x), g(x))$ is, the smaller the error of predicting $s(x)$ should become. Proximity is computed with respect to the relative weight each class carries. Therefore, lower mass accumulation between two classes leads to larger proximity between them.

Setting $prox$ as a closeness metric acknowledges the closeness variability between adjacent classes, recognizing that adjacent class pairs may exhibit different informational distance. To use $prox$ as a distance metric, we propose $dis(c, c') = max_i(prox(c_i, c')) - prox(c, c')$, which maintains the properties of $prox$ while ensuring $dis(c, c') \geq 0$.

Hereinafter, we use $Loss_{prox}$ to indicate that a loss functions $Loss$ makes use of a proximity function for closeness and distance metrics.

Accumulating Cross-Entropy

Equipped with the property desiderata and proximity-aware closeness, we are ready to present SLACE, starting with some basic definitions.

Definition 1 (Dominance Set)**.** *Given a (true) class c^*, the* dominance set of a class c with respect to c^* *is defined to be:*

$$P_{c^*}[c] = \{c' \mid \mathcal{C}(c', c^*) \leq \mathcal{C}(c, c^*),\ c' \in O\} \quad (8)$$

A convenient way to record and present dominance is a binary $K \times K$ dominance matrix P_{c^*}, as follows:

$$P_{c^*}(c', c) = \begin{cases} 1, & \mathcal{C}(c', c^*) \leq \mathcal{C}(c, c^*) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $c, c' \in O$. c is dominant over c' with respect to c^* if $P_{c^*}[c', c] = 1$.

Using the dominance matrix we can define the Accumulating Cross-Entropy loss to be

$$acc\_loss(x, p) = -\sum_{i=1}^{K} \log(\sum_{j=1}^{K} P_{c^*}(c_i, c_j) \cdot p(x)_j) \quad (10)$$

where $g(x) = c^*$ and c_i and c_j are the i-th and j-th classes, respectively.

Essentially, for each class in the sum, this loss function accumulates the probabilities of the class and those closer to the true class. Consequently, the part in the log for further classes is bigger, yielding a smaller $-log$ value (as log is monotone). This implies that a class can never contribute more to the loss than classes closer to the true class. Therefore, the true class is assigned with a value equivalent to CE,

as only the probability of the true class is considered. The furthest class is assigned with 0, considering the sum of all probabilities (1).

Summation is performed by closeness to $g(x)$. For example, consider the data distribution in Figure 3b, but this time, assume that $g(x) = c_2$ and that closeness is computed by plugging per class instance numbers into Eq. 7. The closest class to c_2, which is the true class in this example, is c_3. Given probability distribution p, the corresponding Accumulating Cross-Entropy is $acc\_loss(x, p) = -(log(p_2) + log(p_3 + p_2) + log(p_3 + p_2 + p_1)$.

The accumulating cross-entropy loss has the unique property of not only considering all possible classes but also limiting the effect of a class in the loss to not exceed that of closer classes to the true class.

We can take the weighting of the classes one step further by combining soft labels into the accumulating cross-entropy loss, yielding SLACE:

$$\mathsf{SLACE}(x, p) = -\sum_{i=1}^{K} q(x)_i \log(\sum_{j=1}^{K} P_{c^*}(c_i, c_j) \cdot p(x)_j) \tag{11}$$

where q is defined as in Eq. 3. In this way, we keep the integrity of the properties of the accumulating cross-entropy loss while adding weights to the different classes, ensuring the desired differentiation between classes.

Properties of SLACE

We next present three main properties of SLACE, namely convexity, monotonicity, and balance-sensitivity. From Proposition 1, SLACE also satisfies PSR.

Full proofs for all propositions are provided in an extended version of the paper, available online.[1].

Proposition 2 (Convexity)**.** SLACE *is convex for* $p(d)$

The proof is comes from properties of convexity, as SLACE is a sum of convex functions.

Proposition 3 (Ordinal Monotonicity)**.** *The accumulating loss function and* SLACE *satisfy the Ordinal Monotonicity property (Property 1).*

Proof sketch. Given a confidence transition of ϵ from class c_j to class c_i, which is closer to the true class, we want to show that the loss for p', the probability distribution after the change, is smaller than the loss on the original distribution p. To show it, we separate the loss computation, which sums over the possible classes, to three parts: classes closer than c_i, classes between c_i to c_j (including c_i), and classes equal to or further than c_j. The contribution to the loss from the classes in the first and third categories is the same for p' and p. For the second category, we show that the classes contribution to the loss is smaller in p' compared to p, thereby concluding our proof. □

Proposition 4 (balance-sensitivity)**.** SLACE_{prox} *satisfies the balance-sensitivity property (Property 2).*

[1]https://github.com/inbarnachmani/SLACE

Proof sketch. To prove that $Loss_D(x, p') - Loss_D(x, p)$ is greater than $Loss_{D*}(x, p') - Loss_{D*}(x, p)$ (Property 2), it suffices to prove $q_i^D > q_i^{D^*}$, where q_i^D, $q_i^{D^*}$ are the soft labels for class c_i, computed over class distribution D. By definitions of q_i^D, $q_i^{D^*}$ and $prox$ (Eq. 7), we get that q_i^D and $q_i^{D^*}$ share the same numerator, while $q_i^{D^*}$ has a larger denominator, which leads us to the conclusion that $q_i^D > q_i^{D^*}$. □

Comparative Analysis of Loss Functions

Having shown that SLACE satisfies convexity, ordinal monotonicity (and by Proposition 1 also PSR), and balance-sensitivity, we now turn to analyze existing state-of-the-art loss functions for ordinal regression. A summary of the analysis is given in Table 1. The double separator separates state-of-the art loss functions (above) from the loss functions proposed in this work (below).

Properties → ↓ **Loss**	PSR	Cx	OM	balance-sensitivity
CE	✓	✓	✗	✗
SORD	✗	✓	✗	✗
OLL	✓	✓	✗	✗
acc_loss	✓	✓	✓	✗
SLACE_{prox}	✓	✓	✓	✓

Table 1: Analysis of Ordinal Regression Loss Functions

Multi-Class Cross Entropy

The multi-class cross entropy loss function is given in Eq. 1. Cross entropy is convex (Kawaguchi, Huang, and Kaelbling 2019) and satisfies PSR (Gneiting and Raftery 2007). It is easy to see it does not satisfy ordinal monotonicity or balance-sensitivity, as it neither considers ordering of labels nor the distribution of wrong data labels.

Soft Labels for Ordinal Regression

Cross entropy can be extended to use soft labels, rather than the true class alone, as shown in Eq. 2 (SORD). Using soft labels allows training using all classes and their relative importance through q. For ordinal regression, q can be determined by the distance between classes, as given in Eq. 3.

Kasa et al. (2024) show that SORD is convex and does not satisfy PSR. Moreover, SORD does not satisfy the balance-sensitivity property, as it fails to consider the distribution of the data labels. SORD also fails to satisfy the ordinal monotonicity property for any distance metric. To demonstrate it, consider $\alpha = 1$, a hyper-parameter that is used in Eq. 3. Let $O = \{c_1, c_2, c_3\}$, such that c_1 is closer to c_2 than c_3. Consider the probability distributions for some tuple $x, g(x) = c_1$: $p(x) = (0.1, 0.4, 0.5) \rightarrow s(x) = 3$, $p'(x) = (0.1, 0.89999, 0.00001) \rightarrow s'(x) = 2$ Then,

$$SORD(x, p) \approx 0.79 < SORD(x, p') \approx 1.127$$

Therefore, ordinal monotonicity is not satisfied by SORD.

Ordinal Log-Loss

Castagnos, Mihelich, and Dognin (2022); Polat et al. (2022) presented a method of embedding ordinality into a loss function with a focus on class distance.

$$OLL(x,p) = -\sum_{i=1}^{K} dis(c_i, g(x))^{\alpha} \log(1 - p(x)_i) \quad (12)$$

OLL is convex and satisfies PSR (Kasa et al. 2024). For reasons similar to SORD, OLL does not satisfy the balance-sensitivity property. For ordinal monotonicity, when using the same example as before with $\alpha = 1$, we get

$$OLL(x,p) \approx 0.824 < OLL(x,p') \approx 1$$

Therefore, ordinal monotonicity is not satisfied either.

Experiments

We tested SLACE over multiple ordinal regression benchmark datasets, both inherently ordinal datasets and binned regression datasets, modified for ordinal regression. Following (Gutiérrez et al. 2015; Gutiérrez et al. 2016), we focus on tabular data. Other data modalities are left for future work. The main insights of the empirical analysis can be summarized as follows.

- SLACE$_{prox}$ outperforms state-of-the-art loss functions across all evaluation metrics.
- SLACE$_{prox}$'s dominance is robust to class number.
- Proximity-awareness benefits loss functions using soft labels.

We next describe experimental setup, datasets, evaluation metrics and baselines, before detailing the evaluation results. All datasets and code are available in an online repository.[2]

Experimental Setup

Experiments were performed on a server with 2 Nvidia GeForce RTX 2080 Ti, Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz and a CentOS 7.9 operating system. To compare loss function performance, we use an XGBoost classifier (Chen and Guestrin 2016),[3] configured with the following hyperparameters: colsample_bytree = 0.5, learning_rate = 0.1, max_depth = 10, and n_estimators = 100. Evaluated loss functions where tested using $\alpha \in \{0.3, 0.5, 0.8, 1, 2, 3, 4, 7, 10, 15, 20, 25\}$ (see eqs. 3 and 12). To ensure robustness and for testing statistical significance, we initialized the model with 10 different random seeds and averaged the performance metrics over these runs for each dataset. Statistical significance of the observed differences was evaluated using a paired t-test with p-value < 0.05.

Datasets

The datasets presented by Gutiérrez et al. (2015); Gutiérrez et al. (2016) are commonly used as benchmark datasets for tabular ordinal regression (Bartley, Liu, and Reynolds 2019; Lázaro and Figueiras-Vidal 2023; Gutiérrez, Pérez-Ortiz, and Suárez 2017). We use 10 of the inherently ordinal datasets presented and 9 of the regression datasets. For additional details see supplementary materials and online repositories (Markelle Kelly 2017; Vanschoren et al. 2013; Gutiérrez et al. 2015; Gutiérrez et al. 2016). For the inherently ordinal data we combined classes with less than 10 instances with adjacent classes to allow effective pattern learning. For the regression data we considered binning into 5, 7, and 10 classes for every dataset with similar sized intervals between classes and similar distributions to the original distribution of the regression dataset.

Evaluation Metrics

For the evaluation we use the following seven metrics.

- **Accuracy (Acc)**: Proportion of correctly predicted instances.
- **Mean Absolute Error (MAE)**: Average absolute difference between predicted and actual values.
- **Closeness Evaluation Metric (CEM)**: proximity (Amigo et al. 2020) as a closeness measure.
- **Kendall's Tau** (τ): Rank correlation coefficient (Kendall 1938), comparing predicted and actual rankings.
- **Quadratic Weighted Kappa (QWK)**: agreement between predicted and actual categories, with heavier penalties for larger errors (Brenner and Kliebsch 1996).
- **Macro Accuracy (Ma Acc)**: Averages accuracy across classes, useful for imbalanced datasets.
- **Absolute Mean Absolute Error (AMAE)**: Averages MAE across classes, useful for imbalanced datasets.

For MAE and AMAE, lower values indicate better performance. For the other metrics, higher scores are preferable.

Baselines

We compare SLACE$_{prox}$ against CE (Eq. 1), SORD (Eq. 2), and OLL (Eq. 12). Both SORD and OLL were demonstrated to outperform other state-of-the-art ordinal loss functions (Frank and Hall 2001; Diaz and Marathe 2019; Kasa et al. 2024). For fair comparison, all losses were evaluated on the best preforming α for each metric in each dataset.

	Loss $\rightarrow$	CE	SORD	OLL
ordinal	CEM	100.0% (0.016)	80.0% (0.01)	90.0% (0.019)
	Acc	100.0% (0.018)	70.0% (0.007)	90.0% (0.021)
	Ma Acc	100.0% (0.037)	70.0% (0.028)	90.0% (0.041)
	MAE	100.0% (-0.029)	80.0% (-0.011)	80.0% (-0.003)
	AMAE	100.0% (-0.067)	90.0% (-0.055)	70.0% (-0.028)
regression	CEM	100.0% (0.013)	77.78% (0.007)	74.07% (0.006)
	Acc	100.0% (0.008)	62.96% (0.002)	66.67% (0.003)
	Ma Acc	100.0% (0.037)	92.59% (0.026)	88.89% (0.027)
	MAE	100.0% (-0.038)	59.26% (-0.006)	59.26% (-0.009)
	AMAE	100.0% (-0.127)	100.0% (-0.067)	85.19% (-0.053)

Table 2: % of cases SLACE$_{prox}$ is equal or better performing for each evaluation metric. In parentheses: mean difference

Evaluation Results

Relative performance: Table 2 provides the percentage of datasets where SLACE$_{prox}$ is equal or better performing

[2] https://github.com/inbarnachmani/SLACE

[3] Gradient-boosted decision trees such as XGBoost are considered state-of-the-art for tabular data (McElfresh et al. 2024).

	Metric → / ↓ Loss	Acc↑	MAE↓	CEM ↑	τ↑	QWK↑	Ma ACC↑	AMAE↓
ordinal	CE	0.669	0.431	0.791	0.655	0.667	0.525	0.661
	SORD	0.68	0.414	0.797	0.666	0.674	0.534	0.649
	OLL	0.667	0.405	0.788	0.686	0.7	0.52	0.622
	SLACE$_{prox}$	**0.687**†‡‡	**0.402**†‡	**0.807**†‡‡	**0.687**†‡	**0.702**†‡	**0.562**†‡‡	**0.594**†‡‡
regression	CE	0.599	0.58	0.768	0.701	0.77	0.487	0.842
	SORD	0.605	0.548	0.774	0.73	0.795	0.497	0.782
	OLL	0.604	0.551	0.775	0.726	0.796	0.497	0.767
	SLACE$_{prox}$	**0.607**†‡‡	**0.542**†‡‡	**0.781**†‡‡	**0.731**†‡	**0.813**†‡‡	**0.524**†‡‡	**0.715**†‡‡

(a) Mean Results on Ordinal (top) and Regression Datasets (bottom) with t-test (0.05) Statistical Significance. †-significant from CE, ‡ - significant from SORD, ǂ-significant from OLL. Note that for MAE and AMAE **lower** values are **better**.

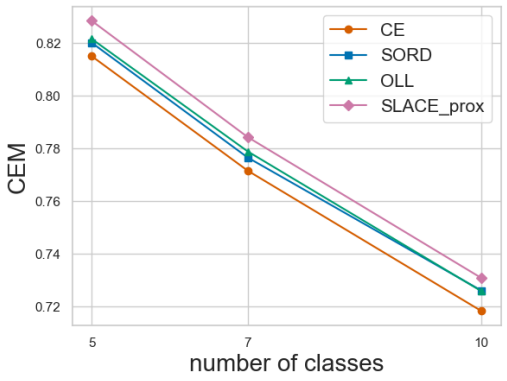


Figure 4: Performance over 5, 7, and 10 classes with binned regression data.

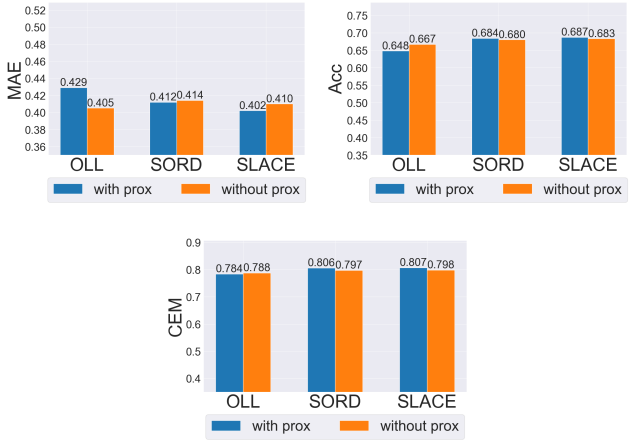


Figure 5: OLL, SORD, and SLACE with and without prox on ordinal datasets. For MAE, lower is better.

than the three other loss functions for each of the evaluation metrics and the mean difference. For all metrics, SLACE$_{prox}$ outperforms state-of-the-art loss functions for most of the evaluated datasets. Specifically, it achieves better results in over 70% of the datasets for CEM, macro accuracy, and AMAE, and in over 59% for accuracy and MAE.

Absolute performance: To compare SLACE against state-of-the-art loss functions, we present the mean results for each metric, separated into ordinal (Table 3a(top)) and regression (Table 3a(bottom)) datasets. Boldface marks the best performing loss function and underlined the second best. The table shows statistically significant results, indicating that SLACE$_{prox}$ consistently outperforms all state-of-the-art loss functions across all metrics. Compared to the top performing baseline (OLL), SLACE improves AMAE by 4.7% and 7.2% and Ma Acc by 5.2% and 5.4% over the ordinal and adjusted regression datasets, respectively.

Robustness: To analyze robustness with respect to number of classes, we use the adjusted (binned) regression datasets, allowing us to analyze the same data with varying number of classes. Figure 4 presents this analysis for the CEM metric over the regression datasets, with 5, 7, and 10 classes. The figure illustrates that while the performance of all loss functions declines as the number of classes increases, SLACE$_{prox}$ performance remains superior.

Impact of proximity-awareness: Figure 5 compares OLL, SORD and SLACE, with and without integrating proximity (prox, Eq. 7) for accuracy (Acc), MAE, and CEM on the ordinal datasets. Our first observation is that loss functions that use soft labels (Eq. 3) generally benefit from integrating proximity-awareness. We also observe that SLACE provides superior performance on both Acc and MAE (where lower is better), either with or without proximity-awareness.

Conclusion

In this work we presented SLACE, a loss function that is both monotonic and balance-sensitive for ordinal regression. SLACE is well-suited to train models in settings where classes are naturally ordinal, with semantics of varying distances between neighboring classes and changing class sizes. We argue and show empirically that SLACE is better suited than state-of-the-art for ordinal regression.

In future work, we shall investigate the impact different class sizes and varying distances have on the performance of models. Also, our empirical analysis focused on tabular data and we intend to extend our investigation to include multiple data types, with emphasis on multi-modal analysis.

Acknowledgements

The work of B. Genossar is supported by the Irwin and Joan Jacobs Excellence Fellowship. The work of R. Shraga was supported in part by NSF award number IIS-2348121. The work of A. Gal is supported in part by J.P. Morgan Fund (2028127) at the Technion.

References

Albuquerque, T.; Cruz, R.; and Cardoso, J. S. 2021. Ordinal losses for classification of cervical cancer risk. *PeerJ Computer Science*, 7: e457.

Alicioglu, G.; Sun, B.; and Ho, S. S. 2020. Assessing Accident Risk using Ordinal Regression and Multinomial Logistic Regression Data Generation. In *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Amigo, E.; Gonzalo, J.; Mizzaro, S.; and Carrillo-de Albornoz, J. 2020. An Effectiveness Metric for Ordinal Classification: Formal Properties and Experimental Results. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Baccianella, S.; Esuli, A.; and Sebastiani, F. 2009. Evaluation measures for ordinal regression. In *Ninth international conference on intelligent systems design and applications*.

Bartley, C.; Liu, W.; and Reynolds, M. 2019. Enhanced Random Forest Algorithms for Partially Monotone Ordinal Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 3224–3231.

Bertinetto, L.; Müller, R.; Tertikas, K.; Samangooei, S.; and Lord, N. A. 2019. Making Better Mistakes: Leveraging Class Hierarchies with Deep Networks. *CoRR*, abs/1912.09393.

Brenner, H.; and Kliebsch, U. 1996. Dependence of weighted kappa coefficients on the number of categories. *Epidemiology*, 199–202.

Castagnos, F.; Mihelich, M.; and Dognin, C. 2022. A Simple Log-based Loss Function for Ordinal Text Classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, 4604–4609. Gyeongju, Republic of Korea: International Committee on Computational Linguistics.

Chawla, N. V.; Japkowicz, N.; and Kotcz, A. 2004. Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1): 1–6.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Collins, E.; Rozanov, N.; and Zhang, B. 2018. Evolutionary Data Measures: Understanding the Difficulty of Text Classification Tasks. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*.

Diaz, R.; and Marathe, A. 2019. Soft Labels for Ordinal Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Domingues, I.; Amorim, J. P.; Abreu, P. H.; Duarte, H.; and Santos, J. 2018. Evaluation of oversampling data balancing techniques in the context of ordinal classification. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Frank, E.; and Hall, M. 2001. A simple approach to ordinal classification. In *European Conference on Machine Learning*, 145–156. Springer.

Gneiting, T.; and Raftery, A. E. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477): 359–378.

Gutiérrez, P. A.; Perez-Ortiz, M.; Sanchez-Monedero, J.; Fernandez-Navarro, F.; and Hervas-Martinez, C. 2015. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1): 127–146.

Gutiérrez, P. A.; Pérez-Ortiz, M.; and Suárez, A. 2017. Class switching ensembles for ordinal regression. In *Advances in Computational Intelligence: 14th International Work-Conference on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14-16, 2017, Proceedings, Part I 14*, 408–419. Springer.

Gutiérrez, P. A.; Pérez-Ortiz, M.; Sánchez-Monedero, J.; Fernández-Navarro, F.; and Hervás-Martínez, C. 2016. Ordinal Regression Methods: Survey and Experimental Study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1): 127–146.

Kasa, S. R.; Goel, A.; Gupta, K.; Roychowdhury, S.; Bhanushali, A.; Pattisapu, N.; and Murthy, P. S. 2024. Exploring Ordinality in Text Classification: A Comparative Study of Explicit and Implicit Techniques. *arXiv preprint arXiv:2405.11775*.

Kawaguchi, K.; Huang, J.; and Kaelbling, L. P. 2019. Every Local Minimum Value Is the Global Minimum Value of Induced Model in Nonconvex Machine Learning. *Neural Computation*, 31(12): 2293–2323.

Kendall, M. 1938. A New Measure of Rank Correlation. *Biometrika*.

Koren, Y.; and Sill, J. 2011. OrdRec: an ordinal model for predicting personalized item rating distributions. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, 117–124. New York, NY, USA: Association for Computing Machinery. ISBN 9781450306836.

Lázaro, M.; and Figueiras-Vidal, A. R. 2023. Neural network for ordinal classification of imbalanced data by minimizing a Bayesian cost. *Pattern Recognition*, 137: 109303.

Liu, Y.; Liu, Y.; Zhong, S.; and Chan, K. C. C. 2011. Semi-supervised manifold ordinal regression for image ranking. In Candan, K. S.; Panchanathan, S.; Prabhakaran, B.; Sundaram, H.; Feng, W.; and Sebe, N., eds., *Proceedings of the 19th International Conference on Multimedia 2011, Scottsdale, AZ, USA, November 28 - December 1, 2011*, 1393–1396. ACM.

Markelle Kelly, K. N., Rachel Longjohn. 2017. UCI Machine Learning Repository.

McElfresh, D.; Khandagale, S.; Valverde, J.; Prasad C, V.; Ramakrishnan, G.; Goldblum, M.; and White, C. 2024. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36.

Merkle, E.; and Steyvers, M. 2013. Choosing a Strictly Proper Scoring Rule. *Decision Analysis*, 10: 292–304.

Polat, G.; Ergenc, I.; Kani, H. T.; Alahdab, Y. O.; Atug, O.; and Temizel, A. 2022. Class Distance Weighted Cross-Entropy Loss for Ulcerative Colitis Severity Estimation. arXiv:2202.05167.

Rennie, J. D.; and Srebro, N. 2005. Loss functions for preference levels: Regression with discrete ordered labels. In *M-PREF@IJCAI, Vol. 1*. Kluwer Norwell, MA.

Saad, S. E.; and Yang, J. 2019. Twitter Sentiment Analysis Based on Ordinal Regression. *IEEE Access*, 7: 163677–163685.

Shi, Y.; Li, W.; and Sha, F. 2016. Metric Learning for Ordinal Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Suresh, V.; and Ong, D. 2021. Not All Negatives are Equal: LabelAware Contrastive Loss for Fine-grained Text Classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4381–4394.

Vanbelle, S.; and Albert, A. 2009. A note on the linearly weighted kappa coefficient for ordinal scales. *Statistical Methodology*, 6(2): 157–163.

Vanschoren, J.; van Rijn, J. N.; Bischl, B.; and Torgo, L. 2013. OpenML: networked science in machine learning. *SIGKDD Explorations*, 15(2): 49–60.

Wang, L.; Xu, S.; Wang, X.; and Zhu, Q. 2021. Addressing Class Imbalance in Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(11): 10165–10173.

Zhang, Y.; and Cheung, Y.-m. 2020. An Ordinal Data Clustering Algorithm with Automated Distance Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 6869–6876.

Full Proofs

Proof of convexity

Proof. From properties that preserve convexity we know that for $q_1, ..., q_K \geq 0$ and $f_1, ..., f_K$ convex $q_1 f_1 + ... + q_K f_K$ is a convex function.
$\mathsf{SLACE}(x,p) = \sum_{i=1}^{K} q(x)_i(-\log(\sum_{j=1}^{K} P_{c^*}(c_i,c_j) \cdot p(x)_j))$ therefore to show that SLACE is convex for the domain $p(x) \in [0,1]^K$ it is enough to show that $\forall i \in \{1,...,K\}, q(x)_i \geq 0$ and $-\log(\sum_{j=1}^{K} P_{c^*}(c_i,c_j) \cdot p(x)_j)$ is convex.

By the definition of q (Eq. 3) it is easy to see that $\forall i \in \{1,...,K\}, q(x)_i \geq 0$ as $\exp(x) > 0$.

Therefore it is left to show that $\forall i \in \{1,...,K\}, -\log(\sum_{j=1}^{K} P_{c^*}(c_i,c_j) \cdot p(x)_j)$ is convex.

Take $i \in \{1, \cdots, K\}$, notice that $P_{c^*}(c_i,c_j)$ is an indicator, therefore we can write $-\log(\sum_{j=1}^{K} P_{c^*}(c_i,c_j) \cdot p(x)_j) = -\log(\sum_{j \in \{1,\cdots,K\}, \text{ s.t } P_{c^*}(c_i,c_j)=1} p(x)_j)$.

For every j, $h(p(x)_j) = p(x)_j$ is a convex function and therefore $g_i(p(x)_1, \cdots, p(x)_K) = \sum_{j \in \{1,\cdots,K\}, \text{ s.t } P_{c^*}(c_i,c_j)=1} h(p(x)_j)$ is convex as a sum of convex functions.

As $-log(x)$ is convex and non-decreasing in the domain and $g_i(p(x)_1, \cdots, p(x)_K)$ is convex, from properties that preserve convexity, the composition $f_i = -log(g_i(p(x)_1, \cdots, p(x)_K)) = -\log(\sum_{j \in \{1,\cdots,K\}, \text{ s.t } P_{c^*}(c_i,c_j)=1} p(x)_j)$ is convex.

□

Proof of monotonicity

Proof. We will look on a probability distribution p and the same distribution after a weight shift p' as defined in property 1, such that $\mathcal{C}(c_i, g(x)) > \mathcal{C}(c_j, g(x))$ and $p'_i = p_i + \epsilon, p'_j = p_j - \epsilon$. We will show that

$$\mathsf{SLACE}(x,p') < \mathsf{SLACE}(x,p) :$$

$\mathsf{SLACE}(x,p') =$
$-\sum_{n=1}^{K} q_n \log(\sum_{m=1}^{K} P_{c^*}(c_n,c_m) \cdot p'_m) \overset{(a)}{=}$
$-\sum_{n;\mathcal{C}(g(x),c_n)>\mathcal{C}(g(x),c_i)} q_n \log(\sum_{m=1}^{K} P_{c^*}(c_n,c_m) \cdot p_m) - \sum_{n;\mathcal{C}(g(x),c_i)\geq\mathcal{C}(g(x),c_n)>\mathcal{C}(g(x),c_j)} q_n \log(\sum_{m=1}^{K} P_{c^*}(c_n,c_m) \cdot p_m + \epsilon) - \sum_{n;\mathcal{C}(g(x),c_n)\leq\mathcal{C}(g(x),c_i)} q_n \log(\sum_{m=1}^{K} P_{c^*}(c_n,c_m) \cdot p_m + \epsilon - \epsilon) \overset{(b)}{<}$
$-\sum_{n;\mathcal{C}(g(x),c_n)>\mathcal{C}(g(x),c_i)} q_n \log(\sum_{m=1}^{K} P_{c^*}(c_n,c_m) \cdot p_m) - \sum_{n;\mathcal{C}(g(x),c_i)\geq\mathcal{C}(g(x),c_n)>\mathcal{C}(g(x),c_j)} q_n \log(\sum_{m=1}^{K} P_{c^*}(c_n,c_m) \cdot p_m) - \sum_{n;\mathcal{C}(g(x),c_n)\leq\mathcal{C}(g(x),c_i)} q_n \log(\sum_{m=1}^{K} P_{c^*}(c_n,c_m) \cdot p_m) =$
$-\sum_{n=1}^{K} q_n \log(\sum_{m=1}^{K} P_{c^*}(c_n,c_m) \cdot p_m) =$
$\mathsf{SLACE}(x,p)$

(a) We sum all of the classes so we can divide the sum to classes closer than c_i to $g(x)$, $(\mathcal{C}(c_n, g(x)) > \mathcal{C}(c_i, g(x)))$, classes between c_i and c_j, $(\mathcal{C}(c_i, g(x)) \geq \mathcal{C}(c_n, g(x)) > \mathcal{C}(c_j, g(x)))$, and classes as fur as c_j and further, $(\mathcal{C}(c_j, g(x)) \geq \mathcal{C}(c_n, g(x)))$.

(b) $q_n > 0$ (Eq. 3), and -log is monotone decreasing so subtracting ϵ from inside the -log increases the equation.

The same proof without the q ($q_n = 1$) holds and proofs monotonicity for acc_loss.

□

Proof of balance-sensitivity

We will prove that for $dis(c_r, c_j) = max_h(prox(c_h, c_j)) - prox(c_r, c_j)$, $\mathcal{C}(c_r, c_j) = prox(c_r, c_j)$, number of classes K, and $g(x) = c_i$, given the conditions in property 2 SLACE satisfies the balance-sensitivity property:

Proof. donate $q_r^D = q(x)_r$ given class distribution D, $\mathsf{SLACE}_D(x,p') - \mathsf{SLACE}_D(x,p) =$

$-\sum_{j=1}^{K} q_j^D log(\sum_{r=1}^{K} P_{c_i}(c_j,c_r)p'_r)$
$+\sum_{j=1}^{K} q_j^D log(\sum_{r=1}^{K} P_{c_i}(c_j,c_r)p_r) =$
$\sum_{j=1}^{K} q_j^D \cdot$
$\left(log(\sum_{r=1}^{K} P_{c_i}(c_j,c_r)p_r) - log(\sum_{r=1}^{K} P_{c_i}(c_j,c_r)p'_r)\right)$

Notice that for every $j \neq i$, $\sum_{r=1}^{K} P_{c_i}(c_j,c_r)p'_r = \sum_{r=1}^{K} P_{c_i}(c_j,c_r)p_r + \epsilon - \epsilon = \sum_{r=1}^{K} P_{c_i}(c_j,c_r)p_r$ as c_l is the closest to c_i after c_i , therefore, $\mathsf{SLACE}_D(x,p') - \mathsf{SLACE}_D(x,p) = q_i^D \left(log(\sum_{r=1}^{K} P_{c_i}(c_i,c_r)p_r) - log(\sum_{r=1}^{K} P_{c_i}(c_i,c_r)p'_r)\right)$ and because the only class that satisfies $P_{c_i}(c_i,c_r) = 1$ is c_i we have,

$q_i^D \left(log(\sum_{r=1}^{K} P_{c_i}(c_i,c_r)p_r) - log(\sum_{r=1}^{K} P_{c_i}(c_i,c_r)p'_r)\right)$
$= q_i^D \left(log(p_i) - log(p'_i)\right) = q_i^D \cdot \left(log(p_i) - log(p_i - \epsilon)\right)$

Similarly,

$\mathsf{SLACE}_{D^*}(x,p') - \mathsf{SLACE}_{D^*}(x,p) = q_i^{D^*} \cdot (log(p_i) - log(p_i - \epsilon))$

log is monotone increasing which means $log(p_i) - log(p_i - \epsilon) > 0$, therefore, it is sufficient to show that $q_i^D > q_i^{D^*}$.

define, $n_j = |c_j|, n = \sum_{j=1}^{K} n_j$.

$q_i^{D^*} = \frac{\exp(-\alpha \cdot (max_h(prox(c_h^*,c_i^*)) - prox(c_i^*,c_i^*)))}{\sum_{j=1}^{K} \exp(-\alpha \cdot (max_h(prox(c_h^*,c_i^*)) - prox(c_j^*,c_i^*)))} =$

$\frac{\exp(-\alpha \cdot max_h(prox(c_h^*,c_i^*)) + \alpha \cdot prox(c_i^*,c_i^*))}{\sum_{j=1}^{K} \exp(-\alpha \cdot max_h(prox(c_h^*,c_i^*)) + \alpha \cdot prox(c_j^*,c_i^*))} \overset{(a)}{=}$

$\frac{\exp(-\alpha \cdot max_h(prox(c_h^*,c_i^*))) \cdot \exp(\alpha \cdot prox(c_i^*,c_i^*))}{\sum_{j=1}^{K} \exp(-\alpha \cdot max_h(prox(c_h^*,c_i^*))) \cdot \exp(\alpha \cdot prox(c_j^*,c_i^*))} \overset{(b)}{=}$

$\frac{\exp(-\alpha \cdot max_h(prox(c_h^*,c_i^*))) \cdot \exp(\alpha \cdot prox(c_i^*,c_i^*))}{\exp(-\alpha \cdot max_h(prox(c_h^*,c_i^*))) \cdot \sum_{j=1}^{K} \exp(\alpha \cdot prox(c_j^*,c_i^*))} =$

$\frac{\exp(\alpha \cdot prox(c_i^*,c_i^*))}{\sum_{j=1}^{K} \exp(\alpha \cdot prox(c_j^*,c_i^*))} \overset{(c)}{=}$

$\frac{\exp(-log((\frac{1}{n+m}(\frac{n_i+m}{2}))^\alpha))}{\sum_{j=1}^{K} \exp(-log((\frac{1}{n+m}(\frac{n_j^*}{2} + \sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*)))^\alpha)} \overset{(d)}{=}$

$\frac{(\frac{n+m}{1}(\frac{2}{n_i+m}))^\alpha}{\sum_{j=1}^{K}(\frac{1}{\frac{1}{2(n+m)}(n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*)})^\alpha} =$

$\frac{(\frac{2(n+m)}{n_i+m})^\alpha}{\sum_{j=1}^{K}(\frac{2(n+m)}{n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*})^\alpha} =$

$\frac{\frac{(2(n+m))^\alpha}{(n_i+m)^\alpha}}{\sum_{j=1} \frac{(2(n+m))^\alpha}{(n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*)^\alpha}} =$

$\frac{(2(n+m))^\alpha \cdot \frac{1}{(n_i+m)^\alpha}}{(2(n+m))^\alpha \cdot \sum_{j=1}^{K} \frac{1}{(n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*)^\alpha}} =$

$\frac{1}{(n_i+m)^\alpha \cdot \sum_{j=1}^{K} \frac{1}{(n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*)^\alpha}} =$

$\frac{1}{\sum_{j=1}^{K} \frac{(n_i+m)^\alpha}{(n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*)^\alpha}} =$

$\frac{1}{\sum_{j=1}^{K}(\frac{n_i+m}{n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*})^\alpha}$

Similarly, $q_i^D = \frac{1}{\sum_{j=1}^{K}(\frac{n_i}{n_j + 2\sum_{r \neq j \text{ s.t } prox(c_r,c_i) \geq prox(c_j,c_i)} n_r})^\alpha}$

(a) $log(x+y) = log(x) + log(y)$.
(b) $-\alpha \cdot max_h(prox(c_h^*, c_i^*))$ is not dependent on j.
(c) definition of prox and $y \cdot log(x) = log(x^y)$.

(c) $-log(x) = log(\frac{1}{x})$.

As $\alpha > 0$, to show that $q_i^D > q_i^{D^*}$ it is enough to show that for every $j \in \{1,\ldots,K\}$, $\frac{n_i}{n_j + 2\sum_{r \neq j \text{ s.t } prox(c_r,c_i) \geq prox(c_j,c_i)} n_r} \leq \frac{n_i+m}{n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*}$ and that for at least one j it is $<$.

Notice that for $j = i$ both sides of the inequality are equal to one,

and for $j \neq i$, $\frac{n_i+m}{n_j^* + 2\sum_{r \neq j \text{ s.t } prox(c_r^*,c_i^*) \geq prox(c_j^*,c_i^*)} n_r^*} = \frac{n_i+m}{n_j + 2\sum_{r \neq j \text{ s.t } prox(c_r,c_i) \geq prox(c_j,c_i)} n_r + 2m}$ as class i will always be closer in prox to i than $j \neq i$ and making class i bigger doesn't effect the ordering of the classes compered to i and the prox dom matrix, and for $r \neq i$, $n_r^* = n_r$.

denote, $2\sum_{r \neq j \text{ s.t } prox(c_r,c_i) \geq prox(c_j,c_i)} n_r = a$, for every $j \neq i$ we get,

$\frac{n_i}{n_j + a} < \frac{n_i+m}{n_j+a+2m} \Leftrightarrow$
$n_i(n_j + a + 2m) < (n_i + m)(n_j + a) \Leftrightarrow$
$n_i \cdot n_j + n_i \cdot a + 2mn_i < n_i \cdot n_j + n_i \cdot a + m(n_j + a) \Leftrightarrow$
$2mn_i < m(n_j + a) \Leftrightarrow$
$2n_i < n_j + 2\sum_{r \neq j, i \text{ s.t } prox(c_r,c_i) \geq prox(c_j,c_i)} n_r + 2n_i$

and as $n_j + 2\sum_{r \neq j, i \text{ s.t } prox(c_r,c_i) \geq prox(c_j,c_i)} n_r > 0$ this expression is always true. And as shown previously we get that $q_i^D > q_i^{D^*}$.

□