A simple log-based loss function for ordinal text classification

Anonymous ACL submission

Abstract

The cross-entropy loss function is widely used and generally considered the default loss function for text classification. When it comes to ordinal text classification where there is an ordinal relationship between labels, the cross-entropy is not optimal as it does not incorporate the ordinal character into its feedback. In this paper, we propose a new simple loss function called ordinal log-loss (OLL). We show that this loss function outperforms state-of-the-art previously introduced losses on four benchmark text classification datasets.

1 Introduction

For many classification tasks, there is an order on the labels of the target variable. In particular, in natural language processing (NLP) when, for example, we are trying to predict the number of stars associated with a review: it is obvious that when the label is 1 star, predicting 2 stars is better than predicting 5 stars. This type of classification is called ordinal classification (or ordinal regression) and many techniques have been developed in recent years around it. Among the most used techniques, the **ordinal binary classification** consists in decomposing the ordering target variable in several binary ones (Frank and Hall, 2001; Allwein et al., 2000). The **threshold methods** treat the target variable (with N classes) as a continuous real-valued variable and $N-1$ thresholds are introduced (Verwaeren et al., 2012; Herbrich et al., 2000). In the **loss-sensitive classification** the loss function is built such that a higher penalty is assigned if the distance between the prediction and the label is higher. Several losses can be used here: mapping the labels $\{C_1; C_2; ...; C_N\}$ into values $\{1; 2; ...; N\}$ and use the mean squared error. The margin loss or the hinge loss can also be extended for ordinal regression (Rennie and Srebro, 2005). The weighted kappa loss (de la Torre et al., 2018), the earth mover's distance (Hou et al., 2016) or the soft labels (Diaz and Marathe, 2019; Bertinetto et al., 2020) are other examples of modified losses used in ordinal classification.

In order to measure the performance of the ordinal regression there are well known metrics such as the off-by-k-accuracy, the mean absolute error, the mean squared error or Kendall Tau for instance (Cardoso and Sousa, 2011; Gaudette and Japkowicz, 2009).

1.1 Specific contribution

The main contribution of this paper is to introduce a new loss named ordinal log-loss (OLL). This loss is easy to use, adapted to ordinal classification and gives more accurate results than classical existing methods in text classification. The idea behind the OLL is to penalize bad predictions instead of rewarding good predictions like the majority of the losses mentioned before do.

In section 2 we introduce the ordinal log-loss. In section 3 we present the experiments, the metrics used and finally the results.

2 Ordinal Log-Loss

2.1 Definition

As explained in the introduction, in ordinal classification tasks, predictions too distant from the labels can be particularly problematic. While the majority of losses introduced in the literature for ordinal classification (Gutiérrez et al., 2015; Bertinetto et al., 2020; Rennie and Srebro, 2005) tend to encourage predictions close to the labels, we introduce a loss which penalises the critical errors (i.e. the predictions that are the most distant from the correct class).

First, for each ordinal classification task, we define a distance matrix that embodies the distances between each label:

$$D = (d(C_i, C_j))_{(i,j)\in[\![1,N]\!]^2} \quad (1)$$

where N is the number of classes, $C = (C_1, ..., C_N)$ are the different classes and $d(C_i, C_j)$ the distance between label C_i and C_j. We denote for the sake of simplicity $d(i, j)$ for $d(C_i, C_j)$ and y for C_y (the label).
Let $P = (p_1, ..., p_N)$ the output probability distribution of a network for a given prediction. By definition, the cross-entropy loss encourages the models to output a high probability for the correct class.

Equivalently, but from the opposite perspective, we wish that the further a prediction is from the true label, the higher the loss should be. With a simple modification of the cross-entropy loss, we can find such a loss, that we introduce as the ordinal log-loss (OLL):

$$\mathcal{L}_{OLL-\alpha}(P, y) = -\sum_{i=1}^{N} \log(1 - p_i) d(y, i)^{\alpha} \quad (2)$$

where α is a strictly positive hyper-parameter. The novelty of this loss lies in the coefficients $-\log(1-p_i)$. In fact, other articles already considered the following loss: $\sum_{i=1}^{N} p_i d(y, i)$ (obtained by replacing $-\log(1 - p_i)$ by p_i and where α is taken equal to 1) (Hou et al., 2016; Kotsiantis and Pintelas, 2004). Nevertheless, for this latter loss, as explained in (Hou et al., 2016), the optimization does not converge to a desired local minimum. Although we have not reported these results in this article, this is indeed what we observed experimentally and what gave us the idea of the OLL. We wanted to penalize classification errors more strongly and since we have the inequality $-\log(1-p_i) \geq p_i$ for all $p_i \in [0, 1[$, OLL satisfies this property.

2.2 Impact of the α parameter

In the expression 2, α is an hyper-parameter that could be interpreted as a penalizing factor: the greater α is, the higher the loss function is when the distance between the output predictions and the labels is high.

3 Experiments and Results

In this section we first introduce the public datasets (section 3.1) and the metrics (section 3.2) used to compare our loss function to existing ones. Then in section 3.3 we present the different results obtained.

3.1 Datasets

To conduct our experiments, we used the SNLI dataset (Bowman et al., 2015) used for tasks such as Recognizing Textual Entailment (RTE). We also use the Amazon Reviews Corpus (Keung et al., 2020), the Yelp Reviews Dataset (Yelp, 2015) and the Stanford Sentiment Treebank for fine grained classification (SST-5) dataset (Socher et al., 2013).

SNLI: Developped by (Bowman et al., 2015), this corpus is a collection of 570k human-written English (including 10k for testing and 10k for validation) pairs of sentences dedicated to the Natural Language Inference (NLI) task. It is composed of three balanced labels: $C = (\text{entailment}, \text{neutral}, \text{contradiction})$. To accelerate the training, we used a random subsample of 250k rows from the training set. The ordinal relationship between the classes is taken into account by using the matrix defined in equation [7] as the distance matrix.

Amazon Reviews: This dataset, published by (Keung et al., 2020), was obtained by gathering customer reviews of product from several categories published on the Amazon marketplace in six different languages. We only kept the reviews written in English and the corresponding star rating (an integer between 1 and 5). It represents a total dataset of 210k samples, including 5k for testing and 5k for validation.

Yelp Reviews: Extracted from the Yelp Dataset Challenge 2015 data (Yelp, 2015), it was first used as a text classification benchmark in (Zhang et al., 2015). It is a balanced dataset composed of 700k samples of reviews (50k for testing) extracted from Yelp, a website hosting crowd-sourced reviews about businesses. Each sample is a (text, 5-star rating) pair. To reduce the time taken for training, a random subsample of the training set of size 200k was used as the training set, and one of size 20k was used for the validation set.

SST-5: Introduced by (Socher et al., 2013), the Stanford Sentiment Treebank (SST) is a corpus with parse trees enabling sentiment analysis. It is composed of 12k sentences extracted from movie reviews and annotated by 3 humans. In the SST fine-grained version (or SST-5), each phrase is labelled as a 5 star rating corresponding to: negative, somewhat negative, neutral, somewhat positive, positive.

3.2 Metrics

In this article we use the classical metrics for ordinal classification (Cardoso and Sousa, 2011; Gaudette and Japkowicz, 2009).

Off-by-*k* Accuracy: In the case of ordinal classification, the Off-by-k Accuracy, or OB_k, is the percentage of total predictions where the index of the predicted label $\hat{y} \in (C_1, ..., C_N)$ and the one from the true label differ from less than k. In our experiments, we assumed that $\forall i \in [\![2, N]\!]$: $d(C_{i-1}, C_i) = 1$ so the OB_k can be formulated as:

$$OB_k = 100 \times \frac{\sum_{s=1}^{S} \mathbb{1}\{d(y_s, \hat{y_s}) \leq k\}}{S} \quad (3)$$

With S being the number of examples.

Mean Absolute Error for Classification: To measure the mean distance between the predicted labels and the true ones, we use the MAE:

$$MAE = \frac{\sum_{s=1}^{S} d(y_s, \hat{y_s})}{S} \quad (4)$$

where d is the distance defined Section 2.1.

Mean Squared Error for Classification: To complete the MAE, we measure the mean squared error:

$$MSE = \frac{\sum_{s=1}^{S} d(y_s, \hat{y_s})^2}{S} \quad (5)$$

Kendall Tau: The Kendall τ (Kendall, 1938) is a measure of rank correlation between two measured quantities. It is defined as :

$$\tau = \frac{\#\{\text{concordant pairs}\} - \#\{\text{discordant pairs}\}}{\binom{S}{2}} \quad (6)$$

where $\forall (i,j) \in [\![1, S]\!]^2, i < j$, if the sort order of (y_i, y_j) and $(\hat{y_i}, \hat{y_j})$ agrees, then $(y_i, \hat{y_i})$ and $(y_j, \hat{y_j})$ are concordant pairs, and discordant pairs otherwise.

<u>Remark</u>: metrics such as the Accuracy or the F_1 score are often used to evaluate models in classification tasks. But in the particular case of ordinal classification, these metrics are not considered relevant as they do not truly outline the performance of a model. Indeed, if 2 models A and B predict the same amount of samples correctly, but model A predicts all the other samples incorrectly with predictions that are really distant to the true labels, while the wrong predictions of model B are labels that are close to the true ones, then models A and B have the same accuracy, but model B should be considered better than model A. Like the accuracy, the multi-class F_1 score does not take into account the distance between classes and is therefore not appropriate for ordinal classification.

3.3 Experimental Results

3.3.1 Model Used

To conduct our experiments, we have trained the **BERT-tiny** model (Turc et al., 2019) on the four datasets listed in section 3.1. The choice of using a smaller version of BERT (Devlin et al., 2018) was made for several reasons. First, having less parameters, this model is a lot faster to train. Secondly, it produces scores lower than bigger models such as **BERT-base**, allowing to better highlight the impact of different loss functions on scores. Finally, being a smaller version of the BERT model, the results provided here are assumed to be generalised to bigger BERT models and other similar Transformers models. We will release our code on Github.

3.4 Distance Matrices

As explained in section 2, each ordinal classification task comes with distance matrix D that reflects the proximity between the different labels. For the SNLI dataset, the ordered labels are $C = (\text{entailment}, \text{neutral}, \text{contradiction})$ while for the other 3 datasets, the ordered labels are $C = (1, 2, 3, 4, 5)$. As mentioned in section 3.2, for any two neighbors labels, we choose a distance of 1 between them. As a result, the distance matrix for the SNLI task is:

$$D = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix} \quad (7)$$

while the one for the 1 to 5 stars rating tasks is :

$$D = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix} \quad (8)$$

3.5 Procedure

For each dataset, we trained the BERT-tiny model with 4 different types of losses: the cross-entropy, the ordinal log-loss (our loss), the weighted kappa loss (de la Torre et al., 2018), and the soft labels loss (Bertinetto et al., 2020). We wanted to compare our loss with these three other losses for the following reasons: the cross entropy loss is a very

Datasets	Batch Size	Num Epochs	Stopping Rate	Weight Decay
Yelp Reviews	1024	100	5	0.01
Amazon Reviews	1024	100	5	0.01
SST-5	1024	2340	117	0.01
SNLI	1024	80	4	0.01

Table 1: Training parameters for each dataset

Loss	OB1	OB2	MAE	MSE	Kendall Tau
CE	5.25	6	4.25	5.75	6.5
OLL-1	3.25	2.67	3	2.25	2.5
OLL-1.5	**1.5**	**1.33**	2.75	**1.5**	**2**
OLL-2	**1.5**	1.67	5.5	2.25	4.25
WKL	6	3.67	7.25	6.5	6
SOFT-2	6.75	6.33	6	6.5	4.5
SOFT-3	5.25	6	3.25	5.75	4.75
SOFT-4	5	5.67	**2**	5.25	3.75

Table 2: Losses mean rank on each metrics

Dataset	Loss	Learn Rate	OB1	OB2	MAE	MSE	Kendall Tau
Yelp reviews	CE	7.5e-5	92.9 ± 0.1	97.7 ± 0.0	0.529 ± 0.001	0.809 ± 0.001	0.713 ± 0.000
	OLL-1	5e-5	92.7 ± 0.0	98.0 ± 0.0	0.536 ± 0.000	0.796 ±0.000	0.712 ± 0.000
	OLL-1.5	1e-4	93.1 ± 0.0	98.4 ± 0.1	0.530 ± 0.003	0.750 ± 0.004	0.718 ± 0.001
	OLL-2	1e-4	93.3 ±0.1	98.6 ± 0.0	0.534 ± 0.003	0.742 ± 0.003	0.716 ± 0.001
	WKL	1e-4	92.1 ± 0.1	98.2 ± 0.1	0.554 ± 0.003	0.814 ±0.011	0.712±0.001
	SOFT-2	7.5e-5	92.6 ± 0.2	97.7 ± 0.1	0.535 ± 0.005	0.826 ± 0.016	0.712±0.001
	SOFT-3	7.5e-5	92.8 ±0.2	97.7 ±0.1	0.532 ±0.003	0.817 ±0.011	0.712±0.001
	SOFT-4	1e-4	92.9 ±0.0	97.9 ±0.0	0.529 ±0.000	0.804 ±0.000	0.714±0.000
Amazon reviews	CE	5e-5	90.9 ± 0.3	97.8 ±0.1	0.578 ±0.004	0.897 ±0.008	0.692±0.002
	OLL-1	5e-5	92.3 ±0.1	98.5 ±0.1	0.570 ±0.001	0.802 ±0.005	0.699±0.001
	OLL-1.5	2.5e-5	92.5 ±0.2	98.6 ±0.0	0.567 ±0.004	0.787 ±0.009	0.701±0.003
	OLL-2	5e-5	92.5 ±0.0	98.6 ±0.0	0.577 ±0.001	0.791±0.002	0.697±0.000
	WKL	5e-5	91.2 ±0.3	98.5 ±0.1	0.591 ±0.008	0.847 ±0.015	0.698±0.003
	SOFT-2	5e-5	90.7 ±0.1	97.9 ±0.1	0.579 ±0.005	0.897 ±0.012	0.695±0.003
	SOFT-3	5e-5	90.8 ±0.2	97.8 ±0.1	0.577 ±0.004	0.899 ±0.012	0.693±0.002
	SOFT-4	5e-5	90.7 ±0.0	97.7 ±0.0	0.577 ±0.000	0.909 ±0.000	0.694±0.000
SST-5	CE	5e-5	85.2 ±0.2	97.0 ±0.2	0.754 ±0.008	1.171 ±0.012	0.533±0.005
	OLL-1	7.5e-5	86.7 ±0.2	98.0 ±0.1	0.738 ±0.002	1.084 ±0.008	0.548±0.003
	OLL-1.5	7.5e-5	86.9 ±0.2	98.0 ±0.1	0.739 ±0.000	1.081 ±0.005	0.544±0.002
	OLL-2	1e-5	86.3 ±0.4	97.7 ±0.2	0.757 ±0.007	1.121 ±0.016	0.531±0.004
	WKL	1e-5	83.8 ±0.7	97.2 ±0.1	0.806 ±0.019	1.259 ±0.038	0.520±0.009
	SOFT-2	2.5e-5	84.8 ±0.6	96.9 ±0.4	0.754 ±0.004	1.186 ±0.031	0.548±0.003
	SOFT-3	1e-5	85.2 ±0.3	97.0 ±0.2	0.748 ±0.001	1.166 ±0.011	0.544±0.005
	SOFT-4	7.5e-5	86.1 ±0.0	97.3 ±0.0	0.738 ±0.000	1.124 ±0.000	0.549±0.000
SNLI	CE	7.5e-5	97.2 ±0.0		0.208 ±0.000	0.264 ±0.001	0.773±0.001
	OLL-1	1e-4	98.3 ±0.0		0.202 ±0.002	0.237 ±0.002	0.786±0.002
	OLL-1.5	7.5e-5	98.3 ±0.0		0.207 ±0.000	0.241 ±0.000	0.781±0.000
	OLL-2	7.5e-5	98.5 ±0.0		0.214 ±0.003	0.244 ±0.004	0.777±0.003
	WKL	1e-4	97.7 ±0.1		0.238 ±0.005	0.283 ±0.006	0.750±0.005
	SOFT-2	2.5e-5	97.3 ±0.1		0.208 ± 0.004	0.261 ±0.006	0.775 ±0.004
	SOFT-3	1e-4	97.4 ±0.0		0.204 ±0.000	0.257 ±0.001	0.779±0.000
	SOFT-4	7.5e-5	97.3 ±0.0		0.205 ±0.000	0.259 ±0.000	0.776±0.000

Table 3: Losses comparisons on 4 datasets: Yelp reviews, Amazon reviews, SST-5 and SNLI

common loss in text classification and the weighted kappa loss and soft labels loss were recently introduced losses and outperformed a significant number of other losses as explained in the previously cited articles.

For the ordinal log-loss, we chose α in $\{1, 1.5, 2\}$, for the soft labels loss, we chose β in $\{2, 3, 4\}$ because it gave us the best results (although in the original paper, the values used for β are higher). For each loss, we trained the model with 5 different learning rates : 10^{-5}, 2.5×10^{-5}, 5×10^{-5}, 7.5×10^{-5} and 10^{-4}. And for each learning rate, the pre-trained model was trained 5 times. Finally, for each dataset, for each loss, we chose the learning rate that gave the best scores in averages for the 5 independent trainings.

3.6 Results

The results of the experiments are shown in table 2 and 3. In table 3, according to the procedure described in section 3.5, for each line in the table we took the average scores for the 5 independent trainings for the given learning rate. We did not display the OB2 score for SNLI because there are only three classes.

We can observe that the OLL gave better results for all the metrics used, although the SOFT loss is performing well too on the MAE metric. Results of the OLL loss vary with the α parameter : while $\alpha \in \{1, 1.5\}$ gives better results on the SNLI and SST-5 datasets, for $\alpha = 2$, the OLL loss is providing good results on the other 2 datasets. Overall, $\alpha = 1.5$ seems to be a good tradeoff.

To have a clearer idea of which losses perform better, we completed the table 2, where each line displays the average rank of the corresponding loss on the 4 datasets, for each metric. Although the SOFT loss with $\beta = 4$ gives interesting results for the MAE and the Kendall Tau, the OLL loss seems to perform better overall. The impact of the α parameter in the OLL loss vary, depending on the dataset and the number of classes, but the table 2 confirmed that $\alpha = 1.5$ is a good trade-off.

3.7 Conclusion

We introduced a simple and novel loss function specially designed for the ordinal classification task. This loss is intuitive and easy to use. We evaluated our method on four benchmark ordinal text classification datasets. Our loss outperforms state-of-the-art comparable and previously introduced losses. We also experimentally find good hyperparameters to use. We believe that those results could be extended to other machine learning tasks in computer vision, speech or structured data for instance.

References

Erin L Allwein, Robert E Schapire, and Yoram Singer. 2000. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, 1(Dec):113–141.

Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. 2020. Making better mistakes: Leveraging class hierarchies with deep networks. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Jaime S Cardoso and Ricardo Sousa. 2011. Measuring the performance of ordinal classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(08):1173–1195.

Jordi de la Torre, Domenec Puig, and Aida Valls. 2018. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognition Letters*, 105:144–154. Machine Learning and Applications in Artificial Intelligence.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Raul Diaz and Amit Marathe. 2019. Soft labels for ordinal regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4738–4747.

Eibe Frank and Mark Hall. 2001. A simple approach to ordinal classification. In *European conference on machine learning*, pages 145–156. Springer.

Lisa Gaudette and Nathalie Japkowicz. 2009. Evaluation methods for ordinal classification. In *Canadian conference on artificial intelligence*, pages 207–210. Springer.

Pedro Antonio Gutiérrez, Maria Perez-Ortiz, Javier Sanchez-Monedero, Francisco Fernandez-Navarro, and Cesar Hervas-Martinez. 2015. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, 28(1):127–146.

R Herbrich, T Graepel, and K Obermayer. 2000. Large-margin thresholded ensembles for ordinal regression: theory and practice. *Advances in Large Margin Classifiers*, pages 115–132.

Le Hou, Chen-Ping Yu, and Dimitris Samaras. 2016. Squared earth mover's distance-based loss for training deep neural networks. *arXiv preprint arXiv:1611.05916*.

M. G. Kendall. 1938. A New Measure Of Rank Correlation. *Biometrika*, 30(1-2):81–93.

Phillip Keung, Yichao Lu, György Szarvas, and Noah A. Smith. 2020. The multilingual amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Sotiris B Kotsiantis and Panagiotis E Pintelas. 2004. A cost sensitive technique for ordinal classification problems. In *Hellenic Conference on Artificial Intelligence*, pages 220–229. Springer.

Jason DM Rennie and Nathan Srebro. 2005. Loss functions for preference levels: Regression with discrete ordered labels. In *Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling*, volume 1. Citeseer.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Jan Verwaeren, Willem Waegeman, and Bernard De Baets. 2012. Learning partial ordinal class memberships with kernel-based proportional odds models. *Computational Statistics & Data Analysis*, 56(4):928–942.

Yelp. 2015. Yelp dataset challenge.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.