Learning by Minimizing the Sum of Ranked Range

Shu Hu[1], Yiming Ying[2], Xin Wang[3], and Siwei Lyu[1]

[1]Department of Computer Science and Engineering, University at Buffalo, SUNY

[2]Department of Mathematics and Statistics, University at Albany, SUNY, [3]CuraCloud Corporation, Seattle, USA

Problem Description

In forming learning objectives, we often need to aggregate a set of individual values to a single numerical value. Such cases occur in the aggregate loss, which combines individual losses of a learning model over each training sample, and in the individual loss for multi-label learning, which combines prediction scores over all class labels.

Data points z_{10} ••• z_2 z_1 $z_i=(x_i, y_i)$

Learning Model: $s_i = s(f(x_i;\theta), y_i)$
- logistic model $s_i = log(1+e^{-y_i f(x_i;\theta)})$
- hinge model $s_i = [1 - y_i f(x_i;\theta)]_+$
- • • •

Individual Losses: s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 (larger values)

How to form a total Loss Function L?

Aggregate Methods
(Aggregate Loss)

Motivation

Majority sub-group / Minority sub-group / Outlier

Individual Losses: s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 (larger values)

Aggregate Loss: Average (Vapnik, 1992): $L_{avg}(\theta) = \frac{1}{n}\sum_{i=1}^n s_i(\theta)$

Limitation: **Insensitive** to minority sub-group (Class imbalance problem)

Majority sub-group / Minority sub-group / Outlier

Individual Losses: s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 (larger values)

Aggregate Loss: Maximum (Shalev-shwartz et al,. 2016): $L_{max}(\theta) = \max_{1\leq i\leq n} s_i(\theta)$

Limitation: **Sensitive** to outliers (Label noise problem)

Majority sub-group / Minority sub-group / Outlier

Individual Losses: s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 (larger values)

Aggregate Loss: Average top k (Fan et al., 2017): $L_{avt-k}(\theta) = \frac{1}{k}\sum_{i=1}^k s_{[i]}(\theta)$

Limitation: It **dilutes but not exclude** the influences of the outliers

SoRR (Sum of Ranked Range)

Individual Losses: s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 (larger values)

Ranked Range [m=2, k=6]

$$L(S(\theta)) = \sum_{i=m+1}^k s_{[i]}(\theta)$$

where, $S(\theta) = \{s_1(\theta), \cdots, s_n(\theta)\}$
$s_i(\theta)$ is an individual loss (convex function of parameter θ)
$s_{[i]}(\theta)$ is the i-th largest individual loss after sorting the elements in S

$$L(S(\theta)) = \sum_{i=m+1}^k s_{[i]}(\theta) = \left[\sum_{i=1}^k s_{[i]}(\theta) - \sum_{i=1}^m s_{[i]}(\theta)\right]$$

Lemma 1 (Fan et al., 2017):
$$\sum_{i=1}^k s_{[i]} = min_{\lambda\in\mathbb{R}}\{k\lambda + \sum_{i=1}^n [s_i - \lambda]_+\}$$

Theorem 1:
$$L(S(\theta)) = \left[\underbrace{\min_{\lambda\in\mathbb{R}}\left\{k\lambda + \sum_{i=1}^n [s_i(\theta)-\lambda]_+\right\}}_{\text{Convex!}} - \underbrace{\min_{\hat\lambda\in\mathbb{R}}\left\{m\hat\lambda + \sum_{i=1}^n [s_i(\theta)-\hat\lambda]_+\right\}}_{\text{Convex!}}\right] \quad (1)$$

Furthermore, $\hat\lambda > \lambda$, when the optimal solution is achieved.

Optimization of SoRR

Background (Thi et al., 2018):
- DC (difference-of-convex) problem
- DC Algorithm (DCA)

We provide an efficient DC (difference-of-convex) algorithm for solving **SoRR**.

Why?
- DCA is a descent method without line search
- DCA converges from an arbitrary initial point and often converges to a global solution
- The natural DC structure of SoRR

To use DCA to optimize SoRR, we need to solve the convex sub-optimization problem

$$\min_\theta \left[\min_\lambda \left\{k\lambda + \sum_{i=1}^n [s_i(\theta)-\lambda]_+\right\} - \theta^T\hat\theta\right].$$

where,

$$\hat\theta \in \sum_{i=1}^n \partial s_i(\theta)\cdot \mathbb{I}_{[s_i(\theta)>s_{[m]}(\theta)]} \quad (3)$$

Algorithm 1: DCA for Minimizing SoRR
Initialization: $\theta^{(0)}$, $\lambda^{(0)}$, η_l, and two hyperparameters k and m
for $t=0,1,...$ **do**
  Compute $\hat\theta^{(t)}$ with Eq.(3)
  for $l=0,1,...$ **do**
    Compute $\theta^{(l+1)}$ and $\lambda^{(l+1)}$ with Eq.(2)
  end
  Update $\theta^{(t+1)} \leftarrow \theta^{(l+1)}$
end

This problem can be solved using a stochastic subgradient method.
- We first randomly sample $s_{i_l}(\theta^{(l)})$ from the collection of $\{s_i(\theta^{(l)})\}_{i=1}^n$
- then perform the following steps:

$$\theta^{(l+1)} \leftarrow \theta^{(l)} - \eta_l\left(\partial s_{i_l}(\theta^{(l)})\cdot\mathbb{I}_{[s_{i_l}(\theta^{(l)})>\lambda^{(l)}]} - \hat\theta^{(t)}\right)$$
$$\lambda^{(l+1)} \leftarrow \lambda^{(l)} - \eta_l\left(k - \mathbb{I}_{[s_{i_l}(\theta^{(l)})>\lambda^{(l)}]}\right) \quad (2)$$

AoRR (Average of Ranked Range)

Individual Losses: s_{10} s_9 s_8 s_7 s_6 s_5 s_4 s_3 s_2 s_1 (larger values)

Average of Ranked Range [m=2, k=6]

AoRR Aggregate Loss: $L(S(\theta)) = \frac{1}{k-m}\sum_{i=m+1}^k s_{[i]}(\theta)$

Generalization

Aggregate Losses		k	m	Formulations
AoRR	Average	n	0	$L_{avg}(S(\theta)) = \frac{1}{n-0}\sum_{i=1}^n s_{[i]}(\theta)$
	Maximum	1	0	$L_{max}(S(\theta)) = \frac{1}{1-0}\sum_{i=1}^1 s_{[i]}(\theta)$
	Average top k	k	0	$L_{avt-k}(S(\theta)) = \frac{1}{k-0}\sum_{i=1}^k s_{[i]}(\theta)$

AoRR Optimization: Algorithm 1

$$L(S(\theta)) = \frac{n}{k-m}\left[\min_\lambda\left\{\frac{k}{n}\lambda + \frac{1}{n}\sum_{i=1}^n [s(y_i f_\theta(x_i))-\lambda]_+\right\} - \min_{\hat\lambda}\left\{\frac{m}{n}\hat\lambda + \frac{1}{n}\sum_{i=1}^n [s(y_i f_\theta(x_i))-\hat\lambda]_+\right\}\right]$$
$$\xrightarrow[n\to\infty]{\frac{k}{n}\to\nu,\frac{m}{n}\to\mu} \frac{n}{k-m}\left[\min_{\lambda\geq 0}\left\{\mathbb{E}[[s(Yf(X))-\lambda]_+] + \nu\lambda\right\} - \min_{\hat\lambda\geq 0}\left\{\mathbb{E}[[s(Yf(X))-\hat\lambda]_+] + \mu\hat\lambda\right\}\right].$$

Assume existence of λ^* and $\hat\lambda^*$

where, $s_i(\theta) = s(y_i f_\theta(x_i))$
$f_\theta(x_i)$ is the parametric predictor
$y_i \in \{\pm 1\}$

$f^* = \arg\inf_f \mathcal{L}(f, \lambda^*, \hat\lambda^*)$
where
$\mathcal{L}(f,\lambda^*,\hat\lambda^*) := \mathbb{E}[[s(Yf(X))-\lambda^*]_+ - [s(Yf(X))-\hat\lambda^*]_+] + (\nu\lambda^* - \mu\hat\lambda^*)$

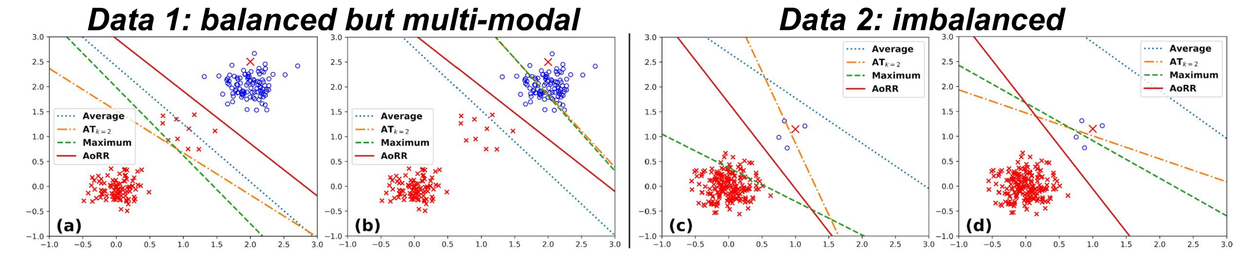
Definition 1 *The AoRR loss is called classification calibrated if there is a minimizer $f^* = \arg\inf_f \mathcal{L}(f,\lambda^*,\hat\lambda^*)$ such as $f^*(x)>0$ if $\eta(x)>1/2$ and $f^*(x)<0$ if $\eta(x)<1/2$.*
where, $\eta(x) = P(Y=1|X=x)$

Theorem 2 *Suppose the individual loss $s:\mathbb{R}\to\mathbb{R}^+$ is non-increasing, convex, differentiable at 0 and $s'(0)<0$. If $0\leq\lambda^*<\hat\lambda^*$, then the AoRR loss is classification calibrated.*

Experiments of AoRR

❖ On synthetic data

Data 1: balanced but multi-modal — Data 2: imbalanced



- (a), (b), (c), and (d) show that the AoRR aggregate loss **outperforms** all other aggregate losses.

❖ On real data

Table 1: Average error rate (%) and standard derivation of different aggregate losses combined with individual logistic loss and hinge loss over 5 datasets. The best results are shown in bold. (R_Max: Robust_Max)

Datasets	Logistic Loss					Hinge Loss				
	Maximum	R_Max	Average	AT_k	AoRR	Maximum	R_Max	Average	AT_k	AoRR
Monk	22.41 (2.95)	21.69 (2.62)	20.46 (2.02)	16.76 (2.29)	**12.69 (2.34)**	22.04 (3.08)	20.61 (3.38)	18.61 (3.16)	17.04 (2.77)	**13.17 (2.13)**
Australian	19.88 (6.64)	17.65 (1.3)	14.27 (3.22)	11.7 (2.82)	**11.42 (1.01)**	19.82 (6.56)	15.88 (1.05)	14.74 (3.10)	12.51 (4.03)	**12.5 (1.55)**
Phoneme	28.67 (0.58)	26.71 (1.4)	25.50 (0.88)	24.17 (0.89)	**21.95 (0.71)**	28.81 (0.62)	24.21 (1.7)	22.88 (1.01)	22.88 (1.01)	**21.95 (0.68)**
Titanic	26.50 (3.35)	24.15 (3.12)	22.77 (0.82)	22.44 (0.84)	**21.69 (0.99)**	25.45 (2.52)	25.08 (1.2)	22.82 (0.74)	22.02 (0.77)	**21.63 (1.05)**
Splice	23.57 (1.93)	23.48 (0.76)	17.25 (0.93)	16.12 (0.97)	**15.59 (0.9)**	23.40 (2.10)	22.82 (2.63)	16.25 (1.12)	16.23 (0.97)	**15.64 (0.89)**

- The AoRR loss achieves the **best performance** on all five datasets.

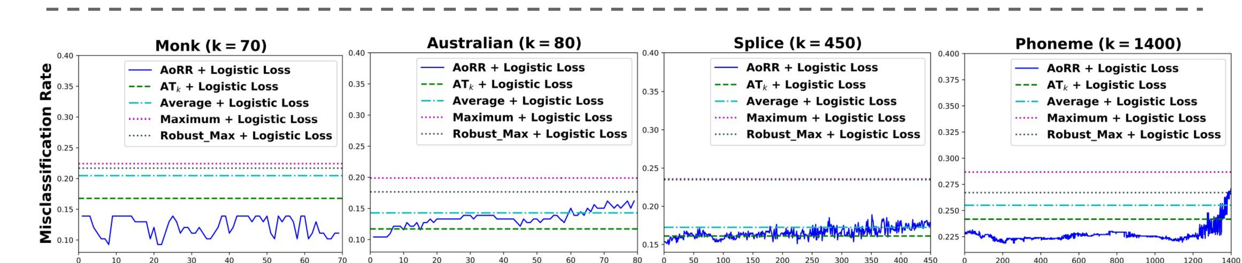


Figure 1: Tendency curves of error rate of learning AoRR loss w.r.t. m on four datasets.

- There is a clear range of m with **better performance** than the corresponding AT_k loss.

TKML (Top k Multi-Label)

In training, the classifier is expected to include as many true labels as possible in the top k outputs.

TKML: $\psi_{k,k+1}(S(\theta)) = s_{[k+1]}(\theta)$
where, $S(\theta) = \{s_j(\theta)\}_{j=1}^l$
$s_j(\theta) = \left[1 + \theta_j^\top x - \min_{y\in Y}\theta_y^\top x\right]_+$

Settings:
A linear predictor $f_\Theta(x) = \Theta^T x$
A set of labels $Y \subset \{1,\cdots,l\}$
Top k prediction scores $\theta_{[1]}^\top x \geq \theta_{[2]}^\top x \geq \cdots \geq \theta_{[k]}^\top x$

TKML $\xrightarrow{\text{Generalization}}$
1. *Conventional multi-class loss (Crammer et al., 2003)* $(|Y| = k = 1)$
2. *Top-k consistent k-guesses multi-class classification loss (Yang et al., 2020)* $(1 = |Y| \leq k < l)$

TKML Optimization: Algorithm 1

Proposition 1 *The TKML loss is a lower-bound to the conventional multi-label loss (Crammer et al., 2003), as*

$$\left[1 + \max_{y\notin Y}\theta_y^\top x - \min_{y\in Y}\theta_y^\top x\right]_+ \geq \psi_{|Y|,|Y|+1}(S(\theta))$$

Experiments of TKML

❖ Multi-label classification

Table 2: Top k multi-label accuracy with its standard derivation (%) on three datasets. The best performance is shown in bold.

Datasets	Methods	k=1	k=2	k=3	k=4	k=5
Emotions	LR	73.54(3.98)	57.48(3.35)	73.20(4.69)	86.60(3.02)	96.46(1.71)
	LSEP	72.18(4.56)	55.85(3.37)	72.18(3.74)	85.58(2.92)	95.85(1.07)
	TKML	**76.80(2.66)**	**62.11(2.85)**	**77.62(2.81)**	**90.14(2.22)**	**96.94(0.63)**
Scene	LR	73.2(0.57)	85.31(0.47)	**94.79(0.79)**	**97.88(0.63)**	**99.7(0.30)**
	LSEP	69.22(3.43)	83.83(4.83)	92.46(4.78)	96.35(3.5)	98.56(1.94)
	TKML	**74.06(0.45)**	**85.36(0.79)**	88.92(1.47)	91.94(0.87)	95.01(0.61)
Yeast	LR	**77.57(0.91)**	**70.59(1.16)**	**52.65(1.23)**	43.26(1.16)	43.49(1.33)
	LSEP	75.5(1.03)	66.84(2.9)	49.72(1.26)	41.90(1.91)	43.01(1.02)
	TKML	76.94(0.49)	67.19(2.79)	45.41(0.71)	**43.47(1.06)**	**44.69(1.14)**

- If we choose the value of k close to the average number of the ground-truth labels per instance, the corresponding classification method **outperforms** the two baseline methods.

❖ Robustness analysis

Table 3: Testing accuracy (%) of two methods on MNIST with different levels of asymmetric noisy labels. The average accuracy and standard deviation of 5 random runs are reported and the best results are shown in bold.

Noise Level	Methods	Top-1 Accuracy	Top-2 Accuracy	Top-3 Accuracy	Top-4 Accuracy	Top-5 Accuracy
0.2	SVM_α	78.33(0.18)	90.66(0.29)	95.12(0.2)	97.28(0.09)	98.49(0.1)
	TKML	**83.06(0.94)**	**94.17(0.19)**	**97.24(0.13)**	**98.47(0.05)**	**99.22(0.01)**
0.3	SVM_α	74.65(0.17)	89.31(0.24)	94.14(0.2)	96.73(0.23)	98.19(0.07)
	TKML	**80.13(1.24)**	**93.37(0.1)**	**96.81(0.22)**	**98.21(0.05)**	**99.08(0.05)**
0.4	SVM_α	68.32(0.32)	86.71(0.42)	93.14(0.49)	96.16(0.32)	97.84(0.18)
	TKML	**75(1.15)**	**92.41(0.14)**	**96.2(0.13)**	**97.95(0.1)**	**98.89(0.04)**

- The gained improvement in performance is getting **more significant** as the level of noise increases.

Conclusion & Future Work

- We introduce a general approach to form learning objectives *SoRR*
- We show that *SoRR* can be optimized with *DC Algorithm*
- We explore two applications
  - *AoRR* aggregate loss for binary classification
  - *TKML* individual loss for multi-label/multiclass classification

In future, we plan to further study the consistency of TKML loss and incorporate SoRR into the learning of deep neural networks.

Code & Datasets

- Code & Datasets can be found at GitHub https://github.com/discovershu/SoRR
- Email: shuhu@buffalo.edu
- This work is supported by NSF research grants (IIS-1816227 and IIS-2008532) as well as an Army Research Office grant (agreement number: W911 NF-18-1-0297)