RankSEG: A Consistent Ranking-based Framework for Segmentation

Ben Dai (The Chinese University of Hong Kong)
Chunlin Li (University of Minnesota)

JMLR ICML International Conference On Machine Learning

Introduction

The primary aim of segmentation is to label each foreground feature/pixel of an input with a corresponding class. Specifically, for a feature vector or an image $\mathbf{X} \in \mathbb{R}^d$, a *segmentation function* $\delta : \mathbb{R}^d \to \{0,1\}^d$ yields a predicted segmentation $\delta(\mathbf{X}) = (\delta_1(\mathbf{X}), \cdots, \delta_d(\mathbf{X}))^\intercal$, where $\delta_j(\mathbf{X})$ represents the predicted segmentation for the j-th feature X_j, and $I(\delta(\mathbf{X})) = \{j : \delta_j(\mathbf{X}) = 1;\ \text{for } j = 1, \cdots, d\}$ is the index set of the segmented features of $\mathbf{X}$ provided by δ.

To access the performance for a segmentation function δ, the **Dice** and **IoU** metrics are introduced and widely used in the literature, both of which measure the overlap between the ground truth and the predicted segmentation:

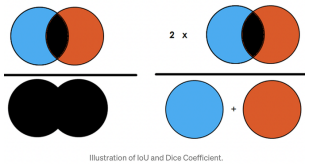


Illustration of IoU and Dice Coefficient.

$$\delta^* = \operatorname{argmax}_\delta \operatorname{Dice}_\gamma(\delta)$$

Mini requirement: any method must be consistent with $\boldsymbol{\delta}^*$; otherwise sub-optimal.

$$\operatorname{Dice}_\gamma(\boldsymbol{\delta}) = \mathbb{E}\Big(\frac{2\big|I(\mathbf{Y}) \cap I(\boldsymbol{\delta}(\mathbf{X}))\big| + \gamma}{\big|I(\mathbf{Y})\big| + \big|I(\boldsymbol{\delta}(\mathbf{X}))\big| + \gamma}\Big) = \mathbb{E}\Big(\frac{2\mathbf{Y}^\intercal\boldsymbol{\delta}(\mathbf{X}) + \gamma}{\|\mathbf{Y}\|_1 + \|\boldsymbol{\delta}(\mathbf{X})\|_1 + \gamma}\Big),$$

$$\operatorname{IoU}_\gamma(\boldsymbol{\delta}) = \mathbb{E}\Big(\frac{\big|I(\mathbf{Y}) \cap I(\boldsymbol{\delta}(\mathbf{X}))\big| + \gamma}{\big|I(\mathbf{Y}) \cup I(\boldsymbol{\delta}(\mathbf{X}))\big| + \gamma}\Big) = \mathbb{E}\Big(\frac{\mathbf{Y}^\intercal\boldsymbol{\delta}(\mathbf{X}) + \gamma}{\|\mathbf{Y}\|_1 + \|\boldsymbol{\delta}(\mathbf{X})\|_1 - \mathbf{Y}^\intercal\boldsymbol{\delta}(\mathbf{X}) + \gamma}\Big),$$

Existing Methods

Most **existing segmentation methods** are developed under a threshold-based framework with two types of loss functions. Given training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1,\cdots,n}$, most existing methods characterize segmentation as a **classification** problem:

$$\widehat{\mathbf{q}} = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{Q}} \frac{1}{n}\sum_{i=1}^n l\big(\mathbf{y}_i, \mathbf{q}(\mathbf{x}_i)\big) + \lambda\|\mathbf{q}\|^2, \qquad \widehat{\boldsymbol{\delta}}(\mathbf{x}) = \mathbf{1}\big(\widehat{\mathbf{q}}(\mathbf{x}) \geq 0.5\big),$$

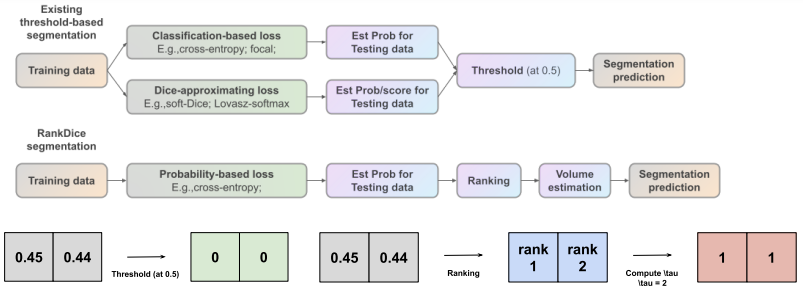
$$\text{(CE)} \quad l_{\text{CE}}\big(\mathbf{y}, \mathbf{q}(\mathbf{x})\big) = -\sum_{j=1}^d \Big(y_j \log\big(\mathbf{q}_j(\mathbf{x})\big) + (1-y_j)\log\big(1-\mathbf{q}_j(\mathbf{x})\big)\Big), \tag{3}$$

$$\text{(Focal)} \quad l_{\text{focal}}\big(\mathbf{y}, \mathbf{q}(\mathbf{x})\big) = -\sum_{i=1}^d \Big(y_j(1-\mathbf{q}_j(\mathbf{x}))^\vartheta \log\big(\mathbf{q}_j(\mathbf{x})\big) + (1-y_j)\mathbf{q}_j^\vartheta(\mathbf{x})\log\big(1-\mathbf{q}_j(\mathbf{x})\big)\Big),$$

The **classification-based framework** yields a **sub-optimal** solution for **segmentation**.

Segmentation is NOT a Classification Problem!

Lemma 9 (*Dai and Li, 2023*). The **classification-based framework** is **not** calibrated when $l(\cdot,\cdot)$ is any classification-calibrated loss, including the cross-entropy loss and the focal loss.



RankSEG framework

Theorem 1 (*Dai and Li, 2023*). *A segmentation rule δ^* is **a global maximizer** of $\operatorname{Dice}_\gamma(\delta)$ iff it satisfies that*

$$\delta_j^*(\mathbf{x}) = \begin{cases} 1 & \text{if } p_j(\mathbf{x}) \text{ ranks top } \tau^*(\mathbf{x}), \\ 0 & \text{otherwise.} \end{cases} \qquad \tau^*(\mathbf{x}) = \operatorname{argmax}_{\tau\in\{0,1,\cdots,d\}} \Big(\sum_{j\in J_\tau(\mathbf{x})} \sum_{l=0}^{d-1} \frac{2p_j(\mathbf{x})\mathbb{P}\big(\Gamma_{-j}(\mathbf{x}) = l\big)}{\tau + l + \gamma + 1} + \sum_{l=0}^{d} \frac{\gamma\mathbb{P}\big(\Gamma(\mathbf{x}) = l\big)}{\tau + l + \gamma} \Big),$$

*$\tau^*(\boldsymbol{x})$ is called **optimal segmentation volume**, where $J_\tau(\mathbf{x})$ is the index set of the τ-largest probabilities, $\Gamma(\mathbf{x}) = \sum_{j=1}^d B_j(\mathbf{x})$, and $\Gamma_{-j}(\mathbf{x}) = \sum_{j'\neq j} B_{j'}(\mathbf{x})$ are Poisson-binomial random variables.*

Obs: both the **Bayes segmentation rule** $\delta^*(\mathbf{x})$ and the **optimal volume** function $\tau^*(\mathbf{x})$ are achievable when the conditional probability $\mathbf{p}(\mathbf{x}) = (p_1(\mathbf{x}), \cdots, p_d(\mathbf{x}))^\intercal$ is well-estimated.

RankSEG inspired by Thm 1

- **Ranking** the conditional probability $p_j(\mathbf{x})$
- searching for the **optimal volume** of the segmented features $\tau(\mathbf{x})$

Step 1 (Conditional probability estimation): Estimate the conditional probability based on logistic regression (the cross-entropy loss):

$$\widehat{\mathbf{q}}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{q}\in\mathcal{Q}} -\sum_{i=1}^n \sum_{j=1}^d \Big(y_{ij}\log\big(q_j(\mathbf{x}_i)\big) + (1-y_{ij})\log\big(1-q_j(\mathbf{x}_i)\big)\Big) + \lambda\|\mathbf{q}\|^2, \tag{5}$$

where $\mathcal{Q}$ is a class of candidate probability functions, $\|\mathbf{q}\|$ is a regularization for a candidate function, and $\lambda > 0$ is a hyperparameter to balance the loss and regularization. For example, $\mathbf{q} \in \mathcal{Q}$ is usually a deep convolutional neural network for image segmentation, and $\|\mathbf{q}\|$ can be a matrix norm of weight matrices in the network.

Step 2 (Ranking): Given a new instance $\mathbf{x}$, rank its estimated conditional probabilities decreasingly, and denote the corresponding indices as $j_1, \cdots, j_d$, that is, $\widehat{q}_{j_1}(\mathbf{x}) \geq \widehat{q}_{j_2}(\mathbf{x}) \geq \cdots \geq \widehat{q}_{j_d}(\mathbf{x})$.

Step 3 (Volume estimation): From (4), we estimate the volume $\widehat{\tau}(\mathbf{x})$ by replacing the true conditional probability $\mathbf{p}(\mathbf{x})$ by the estimated one $\widehat{\mathbf{q}}(\mathbf{x})$:

$$\widehat{\tau}(\mathbf{x}) = \operatorname*{argmax}_{\tau\in\{0,\cdots,d\}} \sum_{s=1}^{\tau}\sum_{l=0}^{d-1} \frac{2}{\tau + l + \gamma + 1}\widehat{q}_{j_s}(\mathbf{x})\mathbb{P}\big(\widehat{\Gamma}_{-j_s}(\mathbf{x}) = l\big) + \sum_{l=0}^{d} \frac{\gamma}{\tau + l + \gamma}\mathbb{P}\big(\widehat{\Gamma}(\mathbf{x}) = l\big), \tag{6}$$

where $\widehat{\Gamma}(\mathbf{x}) = \sum_{j=1}^d \widehat{B}_j(\mathbf{x})$ and $\widehat{\Gamma}_{-j_s}(\mathbf{x}) = \sum_{j\neq j_s} \widehat{B}_j(\mathbf{x})$ are Poisson-binomial random variables, and $\widehat{B}_j(\mathbf{x})$ are independent Bernoulli random variables with success probabilities $\widehat{q}_j(\mathbf{x})$; for $j = 1, \cdots, d$.

Finally, the predicted segmentation $\widehat{\boldsymbol{\delta}}(\mathbf{x}) = (\widehat{\delta}_1(\mathbf{x}), \cdots, \widehat{\delta}_d(\mathbf{x}))^\intercal$ is given by selecting the indices of the top-$\widehat{\tau}(\mathbf{x})$ conditional probabilities:

$$\widehat{\delta}_j(\mathbf{x}) = 1, \text{ if } j \in \{j_1, \cdots, j_{\widehat{\tau}(\mathbf{x})}\}; \quad \widehat{\delta}_j(\mathbf{x}) = 0, \text{ otherwise.} \tag{7}$$

Lemma 10 (*Dai and Li, 2023*). The proposed **RankSEG** is calibrated or consistency.

We propose three algorithms that can solve (5) - (7) with a computational complexity of $O(d\log(d))$.

Completely scalable in practice.

Further theoretical results for the proposed framework including the **excess risk bounds** and **convergence rate**.

Framework	Thold	(Dice, IoU) (×.01)
threshold-based	0.1	(56.80, 28.40)
	0.2	(63.90, 32.00)
	0.3	(65.70, 32.80)
	0.4	(65.60, 32.80)
	0.5	(64.20, 32.10)
	0.6	(62.30, 32.00)
	0.7	(59.30, 29.60)
	0.8	(54.20, 27.10)
	0.9	(43.40, 21.70)
RankDice(our)	—	**(67.10, 33.50)**

Additional numerical results are provided on the **CityScapes** and **Kvasir** datasets.

The improvement is consistently observed.

Model	Loss	Threshold (at 0.5) (mDice, mIoU) (×.01)	Argmax (mDice, mIoU) (×.01)	mRankDice (our) (mDice, mIoU) (×.01)
DeepLab-V3+ (resnet101)	CE	(63.60, 56.70)	(61.90, 55.30)	(64.01, 57.01)
	Focal	(62.70, 55.01)	(60.50, 53.20)	(62.90, 55.10)
	BCE	(63.30, 31.70)	(59.90, 29.90)	**(64.60, 32.30)**
	Soft-Dice	—	—	—
	B-Soft-Dice	—	—	—
	LovaszSoftmax	(57.70, 51.60)	(56.20, 50.30)	(57.80, 51.60)
PSPNet (resnet50)	CE	(64.60, 57.10)	(63.20, 55.90)	(65.40, 57.80)
	Focal	(64.00, 56.10)	(63.90, 56.10)	(66.60, 58.50)
	BCE	(64.20, 32.10)	(65.20, 32.60)	**(67.10, 33.50)**
	Soft-Dice	(59.60, 54.00)	(58.80, 53.20)	(60.00, 54.30)
	B-Soft-Dice	(63.30, 31.60)	(54.00. 27.00)	(64.30, 32.20)
	LovaszSoftmax	(62.00, 55.20)	(60.80, 54.10)	(62.20, 55.40)
FCN8 (resnet101)	CE	(49.50, 41.90)	(45.30, 38.40)	(50.40, 42.70)
	Focal	(50.40, 41.80)	(47.20, 39.30)	(51.50, 42.50)
	BCE	(46.20, 23.10)	(44.20, 22.10)	(47.70, 23.80)
	Soft-Dice	—	—	—
	B-Soft-Dice	—	—	—
	LovaszSoftmax	(39.80, 34.30)	(37.30, 32.20)	(40.00, 34.40)

Table 3: Averaged mDice and mIoU of *threshold*, *argmax*, and the proposed *mRankDice* based on state-of-the-art models/losses on **PASCAL VOC 2012** *val* set. "—" indicates that either the performance is significantly worse or the training is unstable. Gray color indicates that *RankDice*/*mRankDice* is inappropriately applied to a loss function which is not strictly proper. The best performance in each model is bold-faced.