

STOCHASTIC SMOOTHING OF THE TOP-K CALIBRATED HINGE LOSS FOR DEEP IMBALANCED CLASSIFICATION

Joseph Salmon

IMAG, Univ Montpellier, CNRS
Institut Universitaire de France (IUF)



UNIVERSITÉ DE
MONTPELLIER



Inria

COLLABORATION WITH THE PL@NTNET TEAM

FLOWER POWER IN MONTPELLIER



Mainly joint work with:

Camille Garcin (Univ. Montpellier, IMAG)

Maximilien Servajean (Univ. Paul-Valéry-Montpellier, LIRMM, Univ. Montpellier)

Alexis Joly (Inria, LIRMM, Univ. Montpellier)

and:



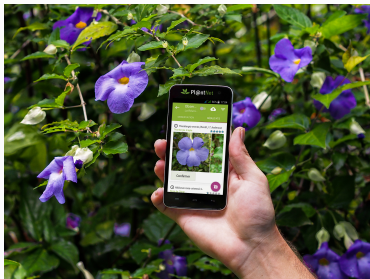
Pierre Bonnet (CIRAD, AMAP)

Antoine Affouard, J-C. Lombardo, Titouan Lorieul, Mathias Chouet (Inria, LIRMM, Univ. Montpellier)

- ▶ C. Garcin, A. Joly, et al. (2021). “Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution”. In: *NeurIPS Datasets and Benchmarks 2021*
- ▶ C. Garcin, M. Servajean, et al. (2022). “Stochastic smoothing of the top-K calibrated hinge loss for deep imbalanced classification”. In: *ICML*

PLANT CLASSIFICATION WITH PL@NTNET

<https://plantnet.org/>



- ▶ ML assisted citizen science
- ▶ > 40,000 species
- ▶ > 10,000,000 annotated images
- ▶ > 1Tb of data \implies Reduction to share with community

← Identification



Résultats



Dipsacus fullonum L.
Cabaret-des-oiseaux

Caprifoliaceae



Valider

4.89

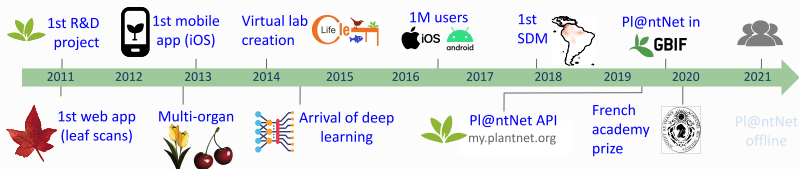


Cichorium intybus L.
Chicorée amère

Asteraceae



Pl@ntNet Key milestones



AGRICULTURAL RESEARCH FOR DEVELOPMENT

Institut de Recherche pour le Développement FRANCE

Supporting agricultural research for sustainable development



Introduction

Pl@ntNet-300K

Dataset characteristics

Dataset construction

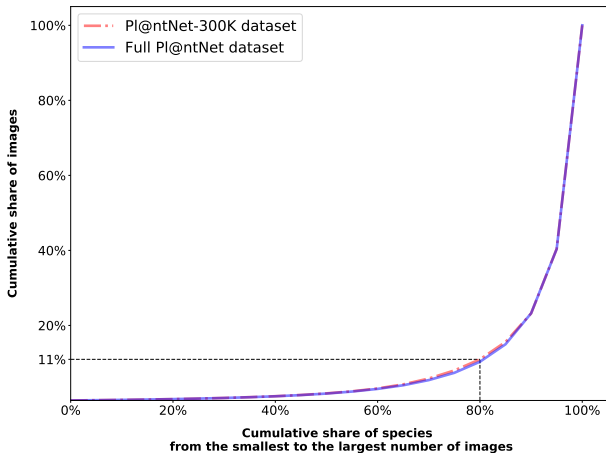
Top-K classification

Experiments

Conclusion

LONG TAILED DISTRIBUTION

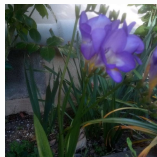
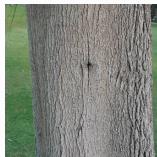
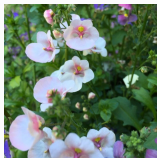
PRESERVED WITH SAMPLING OF GENERA



80% of species account for only 11% of images

INTRA-CLASS VARIABILITY

SAME LABEL/SPECIES BUT VERY DIVERSE IMAGES



*Guizotia
abyssinica*

*Diascia
rigescens*

*Lapageria
rosea*

*Casuarina
cunninghamiana*

*Freesia
alba*

Plant species are challenging to model based on pictures only!

INTER-CLASS AMBIGUITY

DIFFERENT LABELS/SPECIES BUT SIMILAR IMAGES



Cirsium rivulare



Chaerophyllum aromaticum



Conostomium kenysense



Adenostyles leucophylla



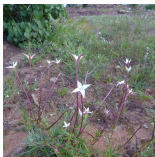
Sedum montanum



Cirsium tuberosum



Chaerophyllum temulum



Conostomium quadrangulare



Adenostyles alliariae



Sedum rupestre

Some species are visually similar (especially within genus)



Introduction

Pl@ntNet-300K

Dataset characteristics

Dataset construction

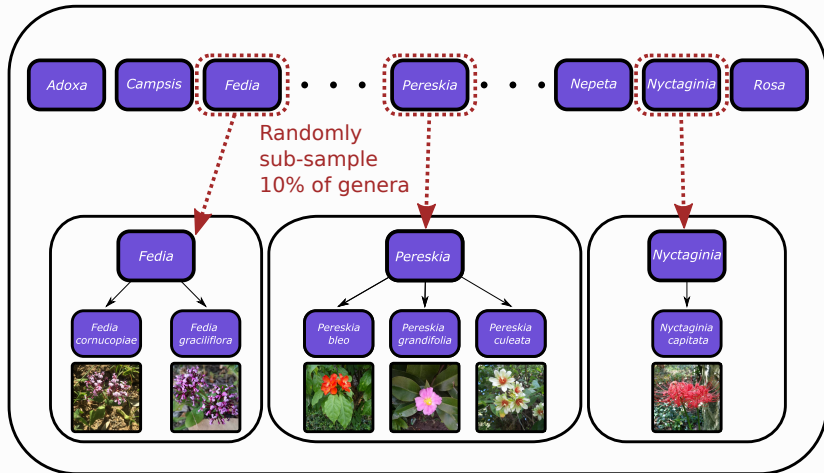
Top-K classification

Experiments

Conclusion

CONSTRUCTION OF PL@NTNET-300K

SUBSAMPLING OF GENERA



Sample at genus level to preserve intra-genus ambiguity



Zenodo, 1 click download

<https://zenodo.org/record/5645731>

Code to train models:

<https://github.com/plantnet/PlantNet-300K>



Introduction

Pl@ntNet-300K

Top-K classification

Motivation

Notation

top-K losses

top-K calibration

top-K smoothing

top-K loss

imbalanced top-K loss


Experiments

Conclusion


LIMITATION OF A SINGLE PROPOSITION



← Identification - Results ▾
World flora




Chaerophyllum aureum L.
Golden-chervil


Confirm  27%

Apiaceae ⓘ

+2




Chaerophyllum bulbosum L.
Turnip-root chervil


Confirm  15%

Apiaceae ⓘ

+2




Conium maculatum L.
Poison hemlock

Confirm  14%

Apiaceae ⓘ

+2



With high class ambiguity, returning a single class is hazardous



Possible solution: return the K "most likely" species for all images

- ▶ Pros for a small K :
ease user experience, handle screen size constraints (think mobile !)

Note: Pl@ntNet suggests species + visual propositions (most similar images to the query), so the user can narrow down the ambiguity

- ▶ Pros for a large K :
ensure the true class lies in the K returned classes

Choice of K :

- ▶ task-dependant, often $K = 3, 5, \dots$ or even larger for challenging tasks
- ▶ considered fixed by the user for the talk (not tuned)



Introduction

Pl@ntNet-300K

Top-K classification

Motivation

Notation

top-K losses

top-K calibration

top-K smoothing

top-K loss

imbalanced top-K loss

Experiments

Conclusion



- ▶ L : number of **classes**, $[L] := \{1, \dots, L\}$, label space
Pl@ntNet-300K: $L = 1\ 081$ species
- ▶ \mathcal{X} : Feature space
Pl@ntNet-300K: $\mathcal{X} = \mathbb{R}^{256 \times 256 \times 3}$
- ▶ $(X_i, Y_i) \in \mathcal{X} \times [L], i = 1, \dots, n$ *i.i.d.* according to \mathbb{P} (unknown)
Pl@ntNet-300K: **306 146** images
- ▶ $K \in [L]$ is a fixed parameter used for top- K
- ▶ **Set-valued classifier**
 $\Gamma : \mathcal{X} \rightarrow 2^{[L]}$; $2^{[L]}$: set of all subsets of $[L]$

Mathematical goal:

minimize the risk $\mathbb{P}(Y \notin \Gamma(X))$ with cardinality constraints on $\Gamma(X)$

Notation:

- ▶ $p_\ell(x) \triangleq \mathbb{P}(Y = \ell | X = x)$: conditional label probability given an input x
- ▶ Decreasing ordering: $p_{(1)}(x) \geq \dots \geq p_{(L)}(x)$,
i.e., (1) is the most likely class for x , (2) the second most likely class, etc.
Below we also use: $p_{(1)}(x) = p_{i_1(x)}(x), \dots, p_{(L)}(x) = p_{i_L(x)}(x)$

- ▶ Top-K classification:

$$\Gamma_{\text{top-K}}^* \in \arg \min_{\Gamma} \mathbb{P}(Y \notin \Gamma(X)) \quad \implies \quad \Gamma_{\text{top-K}}^*(x) = \{i_1(x), \dots, i_K(x)\}$$

s.t. $|\Gamma(x)| \leq K, \forall x \in \mathcal{X}$

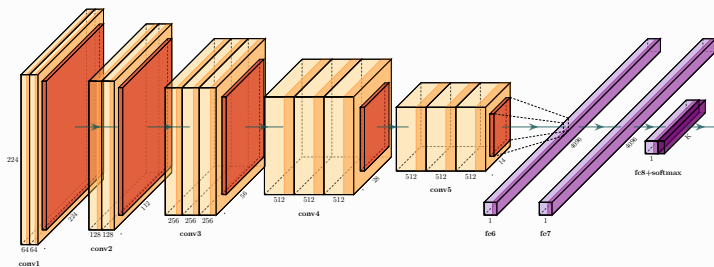
Interpretation:

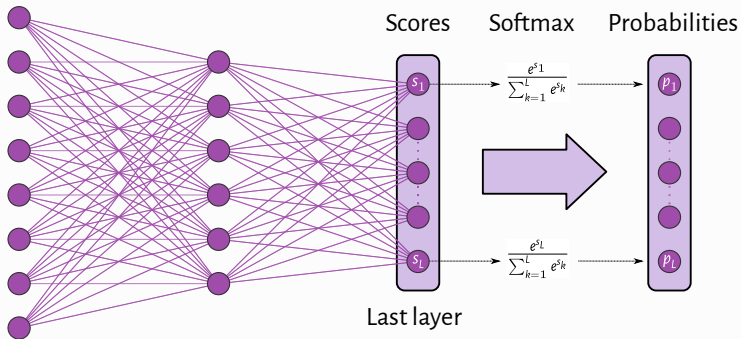
the optimal top-K classifier returns the K most likely classes

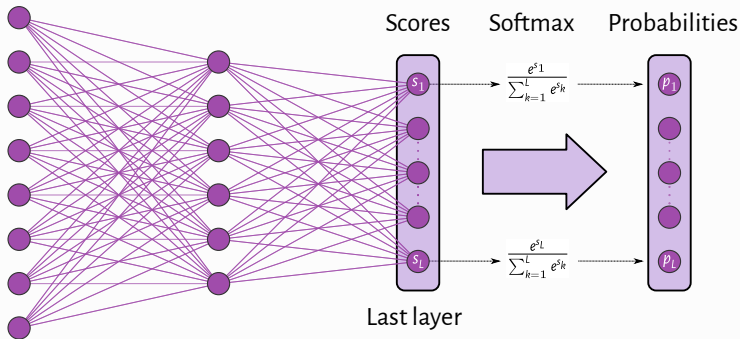
⁽¹⁾ M. Lapin, M. Hein, and B. Schiele (2015). "Top-k multiclass SVM". In: *NeurIPS*, pp. 325–333.

DEEP LEARNING

NOTATION MOSTLY





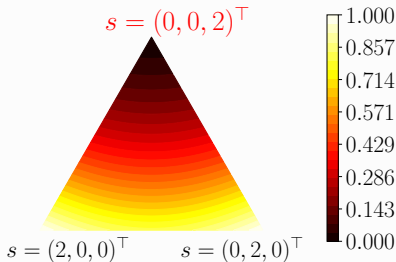


- ▶ From an image, get a score vector $\mathbf{s} = (s_1, \dots, s_L)^T \in \mathbb{R}^L$ (aka logits)
- ▶ s_k : score for class k
- ▶ Reordered scores: $s_{(1)} \geq s_{(2)} \geq \dots \geq s_{(L)}$
- ▶ Prediction: output the "most likely" class, associated to $s_{(1)}$ or $p_{(1)}$

- ▶ Training: cross-entropy (CE) loss + Stochastic Gradient Descent (SGD)

- ▶ $l_{\text{CE}}(\mathbf{s}, y) = -\log\left(\frac{e^{s_y}}{\sum_{k \in [L]} e^{s_k}}\right)$

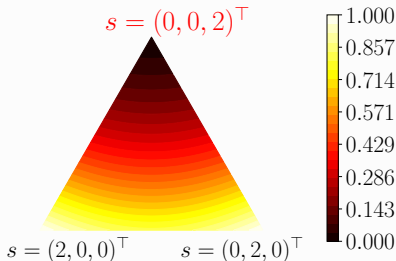
Example : $L = 3, K = 2, y = 3$
(Normalized) level set of $\mathbf{s} \mapsto l_{\text{CE}}(\mathbf{s}, y)$:



- ▶ Training: cross-entropy (CE) loss + Stochastic Gradient Descent (SGD)

- ▶ $l_{\text{CE}}(\mathbf{s}, y) = -\log \left(\frac{e^{s_y}}{\sum_{k \in [L]} e^{s_k}} \right)$

Example : $L = 3, K = 2, y = 3$
(Normalized) level set of $\mathbf{s} \mapsto l_{\text{CE}}(\mathbf{s}, y)$:



- ▶ Not designed to optimize top-K accuracy
- ▶ Can we do better than cross entropy ?

For a score vector $\mathbf{s} \in \mathbb{R}^L$:

Definition

$\text{top}_K : \mathbf{s} \mapsto s_{(K)}$ (K-th largest score)

$\text{top}\Sigma_K : \mathbf{s} \mapsto \sum_{k \in [K]} s_{(k)}$ (sum of K largest scores)

Properties

- ▶ $\nabla \text{top}_K(\mathbf{s}) = \arg \text{top}_K(\mathbf{s}) \in \mathbb{R}^L$:
vector with a single 1 at the K-th largest coordinate of \mathbf{s} , 0 o.w.
- ▶ $\nabla \text{top}\Sigma_K(\mathbf{s}) = \arg \text{top}\Sigma_K(\mathbf{s}) \in \mathbb{R}^L$:
vector with 1's at the K-th largest coordinates of \mathbf{s} , 0 o.w.

⁽²⁾ F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735.



Example on the following score vector: $\mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}$

We have

$$\text{top}_2(\mathbf{s}) = 2.5$$

$$\nabla \text{top}_2(\mathbf{s}) := \arg \text{top}_2(\mathbf{s}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$



Example on the following score vector: $\mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}$

We have

$$\text{top}_2(\mathbf{s}) = 2.5 \qquad \nabla \text{top}_2(\mathbf{s}) := \arg \text{top}_2(\mathbf{s}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\text{top}\Sigma_2(\mathbf{s}) = 4.0 + 2.5 = 6.5 \qquad \nabla \text{top}\Sigma_2(\mathbf{s}) := \arg \text{top}\Sigma_2(\mathbf{s}) = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$



Introduction

Pl@ntNet-300K

Top-K classification

Motivation

Notation

top-K losses

top-K calibration

top-K smoothing

top-K loss

imbalanced top-K loss

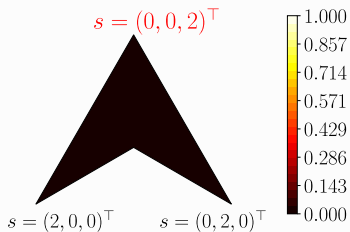
Experiments

Conclusion

Objective: minimize top-K error (0/1 loss):

$$\ell^K(\mathbf{s}, \mathbf{y}) = \mathbb{1}_{\{\text{top}_K(\mathbf{s}) > s_y\}}$$

Problem: piecewise constant function w.r.t. \mathbf{s} , hard to optimize!!!

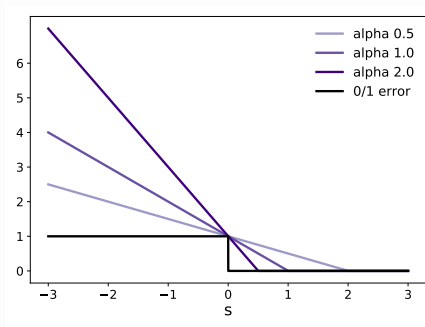


(Normalized) Level sets of $\mathbf{s} \mapsto \ell^K(\mathbf{s}, \mathbf{y})$, $L = 3$, $K = 2$, $y = 3$.

- ▶ Binary case ($L = 2$): $y = 1, y = -1$
- ▶ Score s : predict $y = 1$ if $s > 0, y = -1$ otherwise

Objective: Minimize binary 0/1 error $\ell^{0/1}(s, y) = \mathbb{1}[sy < 0]$.

Upper bound of $\ell^{0/1}$: $\ell^{\text{Hinge}}(s, y) = \alpha \max(0, 1 - \frac{1}{\alpha}sy) = \alpha(1 - \frac{1}{\alpha}sy)_+$



Larger margins ($\frac{1}{\alpha}$) require more confident predictions to achieve a zero loss

Motivation: surrogate top-K loss, similar to hinge loss in binary classification

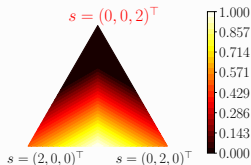
$$\ell_{\text{Hinge}}^K(\mathbf{s}, y) = (1 + \text{top}_K(\mathbf{s}_{\setminus y}) - s_y)_+$$

where $\mathbf{s}_{\setminus y}$ is the vector \mathbf{s} with coordinate y removed

Remark: 1 acts as a *margin* above

Limitations:

- ▶ Experimental: poor performance
- ▶ Theoretical: ℓ_{Hinge}^K is not top-K calibrated (more later)



⁽³⁾ M. Lapin, M. Hein, and B. Schiele (2015). "Top-k multiclass SVM". In: *NeurIPS*, pp. 325–333.



Introduction

Pl@ntNet-300K

Top-K classification

Motivation

Notation

top-K losses

top-K calibration

top-K smoothing

top-K loss

imbalanced top-K loss

Experiments

Conclusion

Question:

When minimizing a surrogate loss ℓ implies minimizing the top-K error ℓ^K ?

Answer: Yes, if ℓ is top-K calibrated

Integrated ℓ -Risk for classifier f

$$\mathcal{R}_\ell(f) \triangleq \mathbb{E}_{(x,y) \sim \mathbb{P}}[\ell(f(x), y)]$$

Integrated Bayes Risk

$$\mathcal{R}_\ell^* \triangleq \inf_{f: \mathcal{X} \rightarrow \mathbb{R}^L} \mathcal{R}_\ell(f)$$

Theorem⁽⁴⁾

Suppose ℓ is top- K calibrated, then, ℓ is top- K consistent, *i.e.*, for any sequence of measurable functions $f^{(n)} : \mathcal{X} \rightarrow \mathbb{R}^L$, we have:

$$\mathcal{R}_\ell \left(f^{(n)} \right) \rightarrow \mathcal{R}_\ell^* \implies \mathcal{R}_{\ell^K} \left(f^{(n)} \right) \rightarrow \mathcal{R}_{\ell^K}^*$$

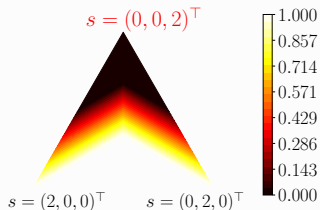
where ℓ^K is the (0/1) top- K loss

Minimizing a top- K calibrated loss implies minimizing the top- K error

⁽⁴⁾ F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735, Theorem 2.2.

A top-K hinge-loss that is top-K calibrated:

$$\ell_{\text{Cal. Hinge}}^k(\mathbf{s}, y) = (1 + \text{top}_{K+1}(\mathbf{s}) - s_y)_+$$



Better theoretical properties, but still fails with deep learning (more later)

Problem: $\mathbf{s} \rightarrow \text{top}_K(\mathbf{s})$ non-smooth and sparse gradient

⁽⁵⁾ F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735.



Introduction

Pl@ntNet-300K

Top-K classification

Motivation

Notation

top-K losses

top-K calibration

top-K smoothing

top-K loss

imbalanced top-K loss

Experiments

Conclusion



Motivation: $\text{top}\Sigma_K$ is a non-smooth, function, smooth it!

- ▶ smoothing parameter $\epsilon > 0$
- ▶ score $\mathbf{s} \in \mathbb{R}^L$

Definition

The ϵ -smoothed version of $\text{top}\Sigma_K$:

$$\text{top}\Sigma_{K,\epsilon}(\mathbf{s}) \triangleq \mathbb{E}_Z[\text{top}\Sigma_K(\mathbf{s} + \epsilon Z)]$$

Z : standard normal random vector, $Z \sim \mathcal{N}(0, \text{Id}_L)$

⁽⁶⁾ Q. Berthet et al. (2020). "Learning with differentiable perturbed optimizers". In: *NeurIPS*.

**Proposition**

For a smoothing parameter $\epsilon > 0$,

- ▶ The function $\text{top}\Sigma_{K,\epsilon} : \mathbb{R}^L \rightarrow \mathbb{R}$ is strictly convex, twice differentiable and \sqrt{K} -Lipschitz continuous.
- ▶ The gradient of $\text{top}\Sigma_{K,\epsilon}$ reads:
$$\nabla_{\mathbf{s}} \text{top}\Sigma_{K,\epsilon}(\mathbf{s}) = \mathbb{E}[\arg \text{top}\Sigma_K(\mathbf{s} + \epsilon Z)]$$
- ▶ $\nabla_{\mathbf{s}} \text{top}\Sigma_{K,\epsilon}$ is $\frac{\sqrt{KL}}{\epsilon}$ -Lipschitz.
- ▶ When $\epsilon \rightarrow 0$, $\text{top}\Sigma_{K,\epsilon}(\mathbf{s}) \rightarrow \text{top}\Sigma_K(\mathbf{s})$.

- ▶ From non-smooth to smooth function with simple stochastic perturbation
- ▶ When $\epsilon \rightarrow 0$, recover the original function



Reminder: $\text{top}_K(\mathbf{s}) \triangleq \text{top}\Sigma_K(\mathbf{s}) - \text{top}\Sigma_{K-1}(\mathbf{s})$

Definition

For any $s \in \mathbb{R}^L$ and $K \in [L]$, the smoothed top-K at level ϵ is:

$$\text{top}_{K,\epsilon}(\mathbf{s}) \triangleq \text{top}\Sigma_{K,\epsilon}(\mathbf{s}) - \text{top}\Sigma_{K-1,\epsilon}(\mathbf{s})$$



Proposition

For a smoothing parameter $\epsilon > 0$,

- ▶ $\text{top}_{P_{K,\epsilon}}$ is $\frac{4\sqrt{KL}}{\epsilon}$ -smooth.
- ▶ For any $\mathbf{s} \in \mathbb{R}^L$, $|\text{top}_{P_{K,\epsilon}}(\mathbf{s}) - \text{top}_K(\mathbf{s})| \leq \epsilon \cdot C_{K,L}$, where $C_{K,L} = K\sqrt{2 \log L}$.

- ▶ Smooth approximation of top_K .
- ▶ Smoothness constant depending on ϵ and problem constants.
- ▶ When $\epsilon \rightarrow 0$, recover initial top-K



Introduction

Pl@ntNet-300K

Top-K classification

Motivation

Notation

top-K losses

top-K calibration

top-K smoothing

top-K loss

imbalanced top-K loss

Experiments

Conclusion



Reminder: $\ell_{\text{Cal. Hinge}}^K(\mathbf{s}, y) = (1 + \text{top}_{K+1}(\mathbf{s}) - s_y)_+$

Definition

We define $\ell_{\text{Noised bal.}}^{K, \epsilon}$ the noised balanced top-K hinge loss as:

$$\ell_{\text{Noised bal.}}^{K, \epsilon}(\mathbf{s}, y) = (1 + \text{top}_{K+1, \epsilon}(\mathbf{s}) - s_y)_+$$

Problem: Untractable: how to deal with the expectation in $\text{top}_{K+1, \epsilon}(\mathbf{s})$?



Solution: Draw B noise vectors Z_1, \dots, Z_B , with $Z_b \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \text{Id}_L)$ for $b \in [B]$.

$$\begin{aligned}\text{top}_{K,\epsilon}(\mathbf{s}) &= \text{top}\Sigma_{K,\epsilon}(\mathbf{s}) - \text{top}\Sigma_{K-1,\epsilon}(\mathbf{s}) \\ &= \mathbb{E}_Z[\text{top}\Sigma_K(\mathbf{s} + \epsilon Z)] - \mathbb{E}_Z[\text{top}\Sigma_{K-1}(\mathbf{s} + \epsilon Z)]\end{aligned}$$

Monte Carlo estimation :

$$\widehat{\text{top}}_{K,\epsilon,B}(\mathbf{s}) = \frac{1}{B} \sum_{b=1}^B \text{top}\Sigma_K(\mathbf{s} + \epsilon Z_b) - \frac{1}{B} \sum_{b=1}^B \text{top}\Sigma_{K-1}(\mathbf{s} + \epsilon Z_b)$$

Easy implementation with deep learning libraries *e.g.*, Pytorch, Tensorflow



Solution: Draw B noise vectors Z_1, \dots, Z_B , with $Z_b \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \text{Id}_L)$ for $b \in [B]$.

$$\begin{aligned}\nabla_{\mathbf{s}} \text{top}_{K, \epsilon}(\mathbf{s}) &= \nabla_{\mathbf{s}} \text{top}_{\Sigma_K, \epsilon}(\mathbf{s}) - \nabla_{\mathbf{s}} \text{top}_{\Sigma_{K-1}, \epsilon}(\mathbf{s}) \\ &= \mathbb{E}[\arg \text{top}_{\Sigma_K}(\mathbf{s} + \epsilon Z)] - \mathbb{E}[\arg \text{top}_{\Sigma_{K-1}}(\mathbf{s} + \epsilon Z)]\end{aligned}$$

Monte Carlo estimation :

$$\widehat{\nabla}_{\text{top}_{K, \epsilon, B}}(\mathbf{s}) = \frac{1}{B} \sum_{b=1}^B \arg \text{top}_{\Sigma_K}(\mathbf{s} + \epsilon Z_b) - \frac{1}{B} \sum_{b=1}^B \arg \text{top}_{\Sigma_{K-1}}(\mathbf{s} + \epsilon Z_b)$$

Easy implementation with deep learning libraries *e.g.*, Pytorch, Tensorflow

$L = 4, K = 2, B = 3, \epsilon = 1.0, \mathbf{s} = \begin{bmatrix} \mathbf{2.4} \\ 2.6 \\ 2.3 \\ 0.5 \end{bmatrix}$. We have $\text{top}_K(\mathbf{s}) = \mathbf{2.4}$ and

$\arg \text{top}_K(\mathbf{s}) = \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \\ 0 \end{bmatrix}$. Assume the three noise vectors sampled are:

$$\mathbf{Z}_1 = \begin{bmatrix} 0.2 \\ -0.1 \\ 0.1 \\ 0.3 \end{bmatrix}, \mathbf{Z}_2 = \begin{bmatrix} 0.1 \\ 0.1 \\ -0.1 \\ 0.1 \end{bmatrix}, \mathbf{Z}_3 = \begin{bmatrix} -0.1 \\ -0.1 \\ 0.1 \\ -0.1 \end{bmatrix}.$$

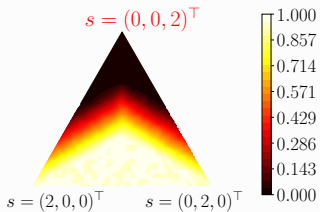
The perturbed vectors are now:

$$\mathbf{s} + \epsilon \mathbf{Z}_1 = \begin{bmatrix} 2.6 \\ \mathbf{2.5} \\ 2.4 \\ 0.8 \end{bmatrix}, \mathbf{s} + \epsilon \mathbf{Z}_2 = \begin{bmatrix} \mathbf{2.5} \\ 2.7 \\ 2.2 \\ 0.6 \end{bmatrix}, \mathbf{s} + \epsilon \mathbf{Z}_3 = \begin{bmatrix} 2.3 \\ 2.5 \\ \mathbf{2.4} \\ 0.4 \end{bmatrix}.$$

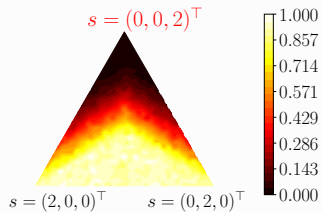
$$\widehat{\text{top}}_{K,\epsilon,B}(\mathbf{s}) = (\mathbf{2.5} + \mathbf{2.5} + \mathbf{2.4})/3 = 2.47,$$

$$\widehat{\nabla \text{top}}_{K,\epsilon,B}(\mathbf{s}) = \frac{1}{3} \left(\begin{bmatrix} 0 \\ \mathbf{1} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathbf{1} \\ 0 \end{bmatrix} \right) = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ 0 \end{bmatrix}.$$

NOISED TOP-K LOSS VISUALIZATION



(a) $\ell^{K, 0.3, 30}$
Noised bal.



(b) $\ell^{K, 1, 30}$
Noised bal.



Introduction

Pl@ntNet-300K

Top-K classification

Motivation

Notation

top-K losses

top-K calibration

top-K smoothing

top-K loss

imbalanced top-K loss

Experiments

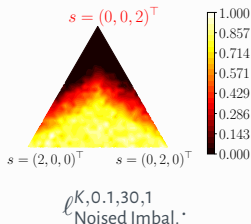
Conclusion

Modification: use larger margins for classes with few examples⁽⁷⁾:

$$\ell_{\text{Noised Imbal.}}^{K, \epsilon, B, m_y}(\mathbf{s}, y) = (m_y + \widehat{\text{top}}_{K+1, \epsilon, B}(\mathbf{s}) - s_y)_+ \quad (1)$$

Set $m_y = C/n_y^{1/4}$, with n_y the number of samples in the training set with class y , and C a hyperparameter to be tuned on a validation set.

Intuition: add more emphasis on rarely seen examples



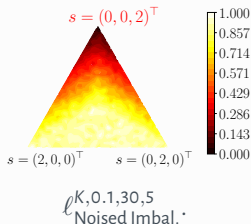
⁽⁷⁾ K. Cao et al. (2019). "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss". In: *NeurIPS*. vol. 32, pp. 1565–1576.

Modification: use larger margins for classes with few examples⁽⁷⁾:

$$\ell_{\text{Noised Imbal.}}^{K, \epsilon, B, m_y}(\mathbf{s}, y) = (m_y + \widehat{\text{top}}_{K+1, \epsilon, B}(\mathbf{s}) - s_y)_+ \quad (1)$$

Set $m_y = C/n_y^{1/4}$, with n_y the number of samples in the training set with class y , and C a hyperparameter to be tuned on a validation set.

Intuition: add more emphasis on rarely seen examples



⁽⁷⁾ K. Cao et al. (2019). "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss". In: *NeurIPS*. vol. 32, pp. 1565–1576.



Introduction

Pl@ntNet-300K

Top-K classification

Experiments

CIFAR100 presentation

Parameter sensitivity

Pl@ntNet-300K results

Conclusion

- ▶ 100 classes, 500 training images per class and 100 test images per class

Superclass

aquatic mammals
fish
flowers
food containers
fruit and vegetables
household electrical devices
household furniture
insects
large carnivores
large man-made outdoor things
large natural outdoor scenes
large omnivores and herbivores
medium-sized mammals
non-insect invertebrates
people
reptiles
small mammals
trees
vehicles 1
vehicles 2

Classes

beaver, dolphin, otter, seal, whale
aquarium fish, flatfish, ray, shark, trout
orchids, poppies, roses, sunflowers, tulips
bottles, bowls, cans, cups, plates
apples, mushrooms, oranges, pears, sweet peppers
clock, computer keyboard, lamp, telephone, television
bed, chair, couch, table, wardrobe
bee, beetle, butterfly, caterpillar, cockroach
bear, leopard, lion, tiger, wolf
bridge, castle, house, road, skyscraper
cloud, forest, mountain, plain, sea
camel, cattle, chimpanzee, elephant, kangaroo
fox, porcupine, possum, raccoon, skunk
crab, lobster, snail, spider, worm
baby, boy, girl, man, woman
crocodile, dinosaur, lizard, snake, turtle
hamster, mouse, rabbit, shrew, squirrel
maple, oak, palm, pine, willow
bicycle, bus, motorcycle, pickup truck, train
lawn-mower, rocket, streetcar, tank, tractor

<https://www.cs.toronto.edu/~kriz/cifar.html>



Introduction

Pl@ntNet-300K

Top-K classification

Experiments

CIFAR100 presentation

Parameter sensitivity

Pl@ntNet-300K results

Conclusion

ϵ	0.0	1e-4	1e-3	1e-2	1e-1	1.0	10.0	100.0
Top-5 acc.	19.38	14.84	11.4	93.36	94.46	94.24	93.78	93.12

CIFAR-100 best validation top-5 accuracy, DenseNet 40-40, $\ell_{\text{Noised bal.}}^{K=5, \epsilon, B=10}$.

- ▶ $\epsilon = 0$ recovers $\ell_{\text{Cal. Hinge}}^K$: bad performance
- ▶ ϵ large enough, relevant coordinates are updated, learning occurs
- ▶ Optimization robust to large values of ϵ



B	1	2	3	5	10	50	100
Top-5 acc	94.28	94.2	94.46	94.52	94.24	94.64	94.52

- ▶ $\ell_{\text{Noised bal.}}^{5,0.2,B}$, CIFAR-100 dataset, DenseNet 40-40 model.
- ▶ B has little influence
- ▶ Using SGD increases the randomness (B noise vectors per batch)
- ▶ In practice set B to a small value *e.g.*, $B = 3$



Introduction

Pl@ntNet-300K

Top-K classification

Experiments

CIFAR100 presentation

Parameter sensitivity

Pl@ntNet-300K results

Conclusion



- ▶ Test set of examples $S_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$
- ▶ $\Gamma_K : \mathcal{X} \rightarrow 2^{[K]}$ learned top-K classifier (model) to evaluate
- ▶ \mathcal{C}_j set of examples of class j : $\mathcal{C}_j = \{\ell \in [L], y_\ell = j\}$

Top-K accuracy: $\frac{1}{n} \sum_{i=1}^n \mathbb{1}[y_i \in \Gamma_K(x_i)]$

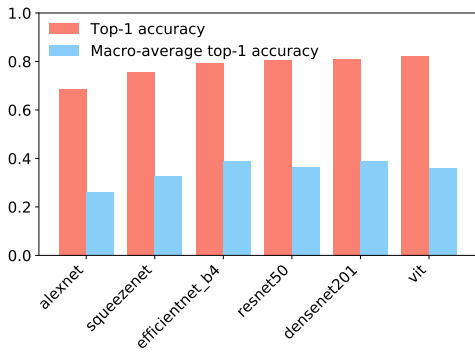
Reflects the performance on classes with lots of examples

Macro-average Top-K accuracy: $\frac{1}{L} \sum_{j=1}^L \frac{1}{|\mathcal{C}_j|} \sum_{\ell \in \mathcal{C}_j} \mathbb{1}[y_\ell \in \Gamma_K(x_\ell)]$

Reflects the performance on all classes regardless of number of examples

CROSS-ENTROPY BASELINE

ACCURACY VS MACRO-AVERAGE ACCURACY



PL@ntNet-300K test *top-1 accuracy* and *macro-average top-1 accuracy* for several neural networks.

Large gap due to long-tailed distribution...



Number of images	Mean bin accuracy
0 – 10	0.09
10 – 50	0.35
50 – 500	0.59
500 – 2000	0.79
> 2000	0.93

Test accuracy depending on number of images per class in training set.
Obtained with ResNet50.

... (many) classes with few examples have low accuracy (hard to learn)



K	ℓ_{CE}	$\ell_{\text{Smoothed Hinge}}^{K, \tau}$	$\ell_{\text{Noised bal.}}^{K, \epsilon, B}$	focal	LDAM	$\ell_{\text{Noised imbal.}}^{K, \epsilon, B, m_y}$
1	35.91	NA	35.44	37.87	40.54	42.36
3	58.91	50.41	59.06	59.96	63.50	64.77
5	69.05	50.71	66.97	69.91	72.23	72.95
10	78.08	46.23	76.08	78.88	80.69	80.85

Macro-average test top-K accuracy on Pl@ntNet-300K, ResNet-50.

- ▶ $\ell_{\text{Smoothed Hinge}}^{K, \tau}$ gives unsatisfactory performance on imbalanced datasets
- ▶ Imbalanced losses fare better than balanced losses
- ▶ Class-wise margin is effective compared to constant margin
- ▶ $\ell_{\text{Noised imbal.}}^{K, \epsilon, B, m_y}$ outperforms other losses on Pl@ntNet-300K



Conclusion

- ▶ A new loss for top- K classification
- ▶ Suitable for training deep learning models
- ▶ Significant performance gains on real databases such as Pl@ntNet (with high ambiguity & a long tail distribution)

Perspectives

- ▶ A fixed set size K is not ideal in practice
 - ▶ Some species are easy to recognize while others are ambiguous
 - ▶ Some images are very informative while others are not
- ▶ Set-valued classification with a varying set size could be more effective

Contact:

✉ `joseph.salmon@umontpellier.fr`








🌐 `http://josephsalmon.eu`

Github: `@josephsalmon`



Twitter: `@salmonjsph`



-  Berrada, L., A. Zisserman, and M. P. Kumar (2018). “Smooth Loss Functions for Deep Top-k Classification”. In: *ICLR*.
-  Berthet, Q. et al. (2020). “Learning with differentiable perturbed optimizers”. In: *NeurIPS*.
-  Cao, K. et al. (2019). “Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss”. In: *NeurIPS*. Vol. 32, pp. 1565–1576.
-  Garcin, C., A. Joly, et al. (2021). “Pl@ntNet-300K: a plant image dataset with high label ambiguity and a long-tailed distribution”. In: *NeurIPS Datasets and Benchmarks 2021*.
-  Garcin, C., M. Servajean, et al. (2022). “Stochastic smoothing of the top-K calibrated hinge loss for deep imbalanced classification”. In: *ICML*.
-  Lapin, M., M. Hein, and B. Schiele (2015). “Top-k multiclass SVM”. In: *NeurIPS*, pp. 325–333.
-  Yang, F. and S. Koyejo (2020). “On the consistency of top-k surrogate losses”. In: *ICML*. Vol. 119, pp. 10727–10735.



- ▶ Reminder: 20 superclasses each containing 5 classes
- ▶ Ex: Super class large carnivores contains the classes "bear", "leopard", "lion", "tiger", "wolf"

For each image in the training set:

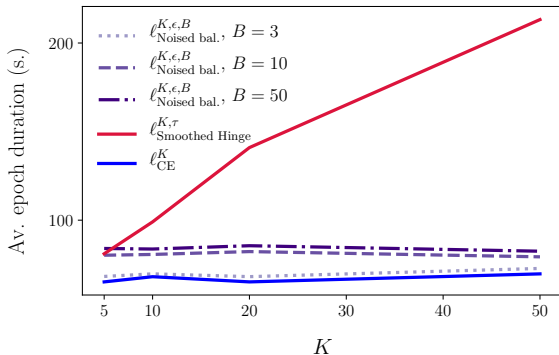
- ▶ With probability p , randomly sample label within the superclass
- ▶ With probability $1 - p$, keep the label unchanged

Possibly wrong class, but same superclass as original dataset.

Label noise p	ℓ_{CE}	$\ell_{\text{Smoothed Hinge}}^{5,1.0}$	$\ell_{\text{Noised bal.}}^{5,0.2,10}$
0.0	94.24	94.34	94.35
0.1	90.39	92.08	92.03
0.2	87.67	90.22	90.68
0.3	85.93	88.82	89.58
0.4	83.74	87.40	87.48

- ▶ CIFAR-100 test Top-5 accuracy, DenseNet 40-40.
- ▶ When $p > 0$, ℓ_{CE} tries to fit corrupted labels while top-K losses merely strives to get the super-class right.
- ▶ $\ell_{\text{Noised bal.}}^{K,\epsilon,B}$ gives good performance and faster to train than $\ell_{\text{Smoothed Hinge}}^{K,\tau}$

Loss: $\ell(\mathbf{s}, y)$	Expression	Param.	Reference
$\ell^K(\mathbf{s}, y)$	$\mathbb{1}_{\{\text{top}_K(\mathbf{s}) > s_y\}}$	K	
$\ell_{\text{CE}}(\mathbf{s}, y)$	$-\ln \left(e^{s_y} / \sum_{k \in [L]} e^{s_k} \right)$	—	
$\ell_{\text{Hinge}}^K(\mathbf{s}, y)$	$(1 + \text{top}_K(\mathbf{s}_{\setminus y}) - s_y)_+$	K	(Lapin, Hein, and Schiele 2015)
$\ell_{\text{CVXHinge}}^K(\mathbf{s}, y)$	$\left(\frac{1}{K} \sum_{k \in [K]} \text{top}_k(\mathbf{1}_L - \delta_y + \mathbf{s}) - s_y \right)_+$	K	(Lapin, Hein, and Schiele 2015)
$\ell_{\text{Cal. Hinge}}^K(\mathbf{s}, y)$	$(1 + \text{top}_{K+1}(\mathbf{s}) - s_y)_+$	K	(Yang and Koyejo 2020)
$\ell_{\text{Smoothed Hinge}}^{K, \tau}(\mathbf{s}, y)$	$\tau \ln \left[\sum_{A \subset [L], A =K} e^{\frac{\mathbb{1}_{\{y \notin A\}}}{\tau} + \sum_{j \in A} \frac{s_j}{K\tau}} \right] - \tau \ln \left[\sum_{A \subset [L], A =K} e^{\sum_{j \in A} \frac{s_j}{K\tau}} \right]$	K, τ	(Berrada, Zisserman, and Kumar 2018)
$\ell_{\text{Noised bal}}^{K, \epsilon, B}(\mathbf{s}, y)$	$(1 + \widehat{\text{top}}_{K+1, \epsilon, B}(\mathbf{s}) - s_y)_+$	K, ϵ, B	proposed
$\ell_{\text{Noised Imbal}}^{K, \epsilon, B, m_y}(\mathbf{s}, y)$	$(m_y + \widehat{\text{top}}_{K+1, \epsilon, B}(\mathbf{s}) - s_y)_+$	K, ϵ, B, m_y	proposed



- ▶ CIFAR-100 dataset, DenseNet 40-40 model
- ▶ $\ell_{\text{Noised bal.}}^{K,\epsilon,B}$ insensitive to K unlike $\ell_{\text{Smoothed Hinge}}^{K,\tau}$

Proposition

For a smoothing parameter $\epsilon > 0$ and a label $y \in [L]$:

- $\ell_{\text{Noised bal.}}^{K, \epsilon}(\cdot, y)$ is continuous and differentiable almost everywhere
- The gradient of $\ell(\cdot, y) \triangleq \ell_{\text{Noised bal.}}^{K, \epsilon}(\cdot, y)$ is given by:

$$\nabla \ell(\mathbf{s}, y) = \mathbb{1}_{\{1 + \text{top}_{K+1, \epsilon}(\mathbf{s}) \geq s_y\}} \cdot (\nabla \text{top}_{K+1, \epsilon}(\mathbf{s}) - \delta_y),$$

where $\delta_y \in \mathbb{R}^L$ is the vector with 1 at coordinate y and 0 elsewhere.

$\Delta_L \triangleq \{\boldsymbol{\pi} \in \mathbb{R}^L : \sum_{k \in [L]} \pi_k = 1, \pi_k \geq 0\}$: probability simplex of size L

Risks

- ▶ Conditional risk: for $x \in \mathcal{X}$, $\boldsymbol{\pi} \in \Delta_L$, $\mathcal{R}_{\ell|x}(\mathbf{s}, \boldsymbol{\pi}) = \mathbb{E}_{y|x \sim \boldsymbol{\pi}}(\ell(\mathbf{s}, y))$
- ▶ Integrated risk for a scoring function f : $\mathcal{R}_{\ell}(f) \triangleq \mathbb{E}_{(x,y) \sim \mathbb{P}}[\ell(f(x), y)]$

Bayes risks :

$$\mathcal{R}_{\ell|x}^*(\boldsymbol{\pi}) \triangleq \inf_{\mathbf{s} \in \mathbb{R}^L} \mathcal{R}_{\ell|x}(\mathbf{s}, \boldsymbol{\pi})$$
$$\mathcal{R}_{\ell}^* \triangleq \inf_{f: \mathcal{X} \rightarrow \mathbb{R}^L} \mathcal{R}_{\ell}(f)$$

Definition⁽⁸⁾

For a fixed $K \in [L]$, and given $\mathbf{s} \in \mathbb{R}^L$ and $\tilde{\mathbf{s}} \in \mathbb{R}^L$, we say that \mathbf{s} is top- K preserving w.r.t. $\tilde{\mathbf{s}}$, denoted $P_K(\mathbf{s}, \tilde{\mathbf{s}})$, if for all $k \in [L]$,

$$\tilde{s}_k > \text{top}_{K+1}(\tilde{\mathbf{s}}) \implies s_k > \text{top}_{K+1}(\mathbf{s})$$

$$\tilde{s}_k < \text{top}_K(\tilde{\mathbf{s}}) \implies s_k < \text{top}_K(\mathbf{s})$$

The negation of this statement is $\neg P_k(\mathbf{s}, \tilde{\mathbf{s}})$.

Roughly speaking: the top- K coordinates of the two vectors are the same

⁽⁸⁾ F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735, Definition 2.3.

**Example:**

- ▶ Consider the vectors $\mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}$ and $\tilde{\mathbf{s}}_1 = \begin{bmatrix} 5.0 \\ 1.0 \\ 6.0 \\ 3.0 \end{bmatrix}$.

\mathbf{s} is top-2 preserving with respect to $\tilde{\mathbf{s}}_1$ because it preserves its top-2 components (the first and third components).

- ▶ Consider the vectors $\mathbf{s} = \begin{bmatrix} 4.0 \\ -1.5 \\ 2.5 \\ 1.0 \end{bmatrix}$ and $\tilde{\mathbf{s}}_2 = \begin{bmatrix} 5.0 \\ 5.5 \\ -1.0 \\ 3.0 \end{bmatrix}$.

\mathbf{s} is not top-2 preserving with respect to $\tilde{\mathbf{s}}_2$ because it changes its top-2 components.

Definition⁽⁹⁾

A loss $\ell : \mathbb{R}^L \times \mathcal{Y} \rightarrow \mathbb{R}$ is top- K calibrated if for all $\pi \in \Delta_L$ and $x \in \mathcal{X}$:

$$\inf_{\mathbf{s} \in \mathbb{R}^L: \neg P_k(\mathbf{s}, \pi)} \mathcal{R}_{\ell|x}(\mathbf{s}, \pi) > \mathcal{R}_{\ell|x}^*(\pi)$$

Interpretation:

ℓ is top- K calibrated if the Bayes risk can only be attained among top- K preserving vectors w.r.t. the conditional probability distribution

⁽⁹⁾ F. Yang and S. Koyejo (2020). "On the consistency of top-k surrogate losses". In: *ICML*. vol. 119, pp. 10727–10735, Definition 2.4.