RiskyZoo: A Library for Risk-Sensitive Supervised Learning

William Wong[1], Audrey Huang[2], Liu Leqi[1], Kamyar Azizzadenesheli[3], Zachary C. Lipton[1]

[1]Carnegie Mellon University, [2]University of Illinois Urbana-Champaign, [3]Purdue University

Contributions

Motivation: While risk functionals and learning procedures have been proposed to assist responsible decision making, their implementations are either **nonexistent** or in **scattered** repositories.

Contributions: We introduce **RiskyZoo** a library which contains standardized implementations of risk-sensitive learning objectives, optimization procedures, and baseline datasets.

RiskyZoo:

- **simplifies** risk-sensitive learning research.
- can be **easily incorporated** into existing PyTorch pipelines.
- is **extensible** to novel objectives, optimizers, and datasets.

Risk Sensitive Learning

Empirical loss minimization may **fail to account for real-world dynamics** such as distribution shifts and desiderata such as robustness.

Risk sensitive learning proposes to **optimize other risk functionals** of the loss distribution. Given i.i.d. training data $\{X_i, Y_i\}_{i=1}^n$ from $\mathbb{P}_{train}$, a loss function l_f, a risk functional ρ that maps a random variable to a real value, and a hypothesis class $\mathcal{F}$, we aim to find:

$$f^*(\rho, \mathbb{P}_{train}) \in \min_{f \in \mathcal{F}} \rho(l_f(X, Y)).$$

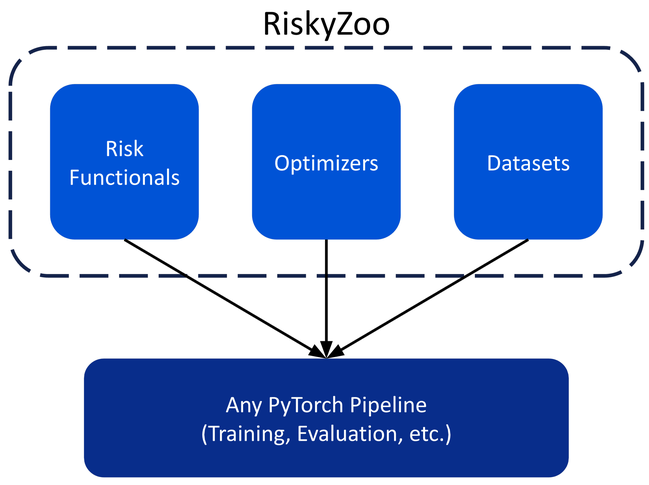
Common risk functionals in literature:

- Expected value: $\rho_{\mathbb{E}}(l_f(X,Y)) = \mathbb{E}[l_f(X,Y)]$
- CVaR: $\rho_{CVaR}\left(l_f(X,Y)\right) = \mathbb{E}\left[l_f(X,Y) \middle| l_f(X,Y) \geq VaR_\alpha\left(l_f(X,Y)\right)\right], \alpha \in [0,1]$
- Entropic Risk: $\rho_{\mathbb{E}nt}(l_f(X,Y)) = \frac{1}{t}\log\mathbb{E}[e^{t l_f(X,Y)}]$
- Human-Aligned Risk: $\rho_H\left(l_f(X,Y)\right) = \mathbb{E}\left[l_f(X,Y) w\left(\mathrm{CDF}\left(l_f(X,Y)\right)\right)\right]$
- Inverted CVaR: $\rho_{\overline{CVaR}}\left(l_f(X,Y)\right) = \mathbb{E}\left[l_f(X,Y) \middle| l_f(X,Y) \leq VaR_\alpha\left(l_f(X,Y)\right)\right]$
- Mean-Variance: $\rho_{MV}\left(l_f(X,Y)\right) = \mathbb{E}\left[l_f(X,Y)\right] + c \cdot \mathrm{Variance}[l_f(X,Y)]$
- Trimmed Risk: $\rho_{Trim}\left(l_f(X,Y)\right) = \mathbb{E}\left[l_f(X,Y) | \mathrm{CDF}(l_f(X,Y)) \in (a, 1-a)\right]$

Algorithmic Properties:

- Risk sensitive models can deal with **distributions shifts** such as covariate shift, label shift such as noisy labels, and provide robustness to **outliers and heavy tails.**

Library Overview



Library Code: github.com/w07wong/RiskyZoo.

RiskyZoo consists of three modules (i) Risk Functionals, (ii) Optimizers, and (iii) Datasets.

Each module can be easily added into existing PyTorch pipelines for tasks such as image classification, risk prediction, etc.

Module: Risk Functionals
All risk functionals described previously are implemented with configurable parameters.

Module: Optimizers
The library's standard optimization procedures are first-order methods[1]. We also implement specialized optimizers for CVaR. Optimization methods can be used in PyTorch training pipelines in place of optimizers such as SGD.

Module: Datasets
We introduce five standardized datasets for risk-sensitive learning research:

1. Classification: Covariate shift
2. Classification: Noisy labels with label shift
3. Classification: Noisy labels without label shift
4. Regression: Fairness
5. Regression: Label Shift

Learning With RiskyZoo

```
...

criterion = torch.nn.CrossEntropyLoss(reduction='none')
risk_functional = riskyzoo.supervised_learning.risk_functionals.CVaR(a=0.2)

pred = model(X)
loss = criterion(pred, ground_truth)
loss = risk_functional(loss)

loss.backward()

...
```

Training risk-sensitive models is as easy as adding two lines.

RiskyZoo In Action

ImageNet Risk Assessment: We conduct risk assessments of pretrained PyTorch ImageNet classifiers by evaluating model performance under each risk functional. While all models achieve similar validation accuracies, Inception has 2x higher mean-variance. When predictability is important, Inception may not be as well suited as other models.

Model	Validation Accuracy	ρ_{MV}
GoogLeNet	0.70	**4.38**
Inception	0.70	14.42
ResNet-18	0.70	5.35
ShuffleNet	0.69	6.72
VGG-11	0.69	5.21

CIFAR-10: We uniformly at random corrupt 80% of CIFAR-10's training labels. The test set remains unchanged. VGG-11 models are learned under each risk functional. Entropic risk, human-aligned risk, and mean variance achieve the highest test accuracies.

Training Objective	Training Accuracy	Test Accuracy
$\rho_{\mathbb{E}}$	0.19	0.47
$\rho_{\mathbb{E}nt}$	**0.21**	**0.52**
ρ_H	0.20	0.51
ρ_{MV}	0.20	0.48

References

[1] Leqi, L., Huang, A., Lipton, Z. C., and Azizzadenesheli, K. Supervised learning with general risk functionals. In *International Conference on Machine Learning*, 2022.