

Class Weighted Classification: Trade-offs and Robust Approaches

Ziyu Xu (Neil), Chen Dan, Justin Khim, Pradeep Ravikumar

Machine Learning Department, Computer Science Department
Carnegie Mellon University
ICML 2020 (July 12th, 2020)



Problem

We look at the class imbalance problem in machine learning, which comes up in applications such as e-commerce, object detection etc.

Contributions

- Fundamental trade-off for different weightings
- Formulation for robust risk on a set of weightings
- Stochastic programming solution to robust risk
- Statistical guarantees for generalization of robust risk (paper)

Organization

- Motivation and previous approaches
- Fundamental trade-off for different weightings
- Formulation for robust risk on a set of weightings
- Stochastic programming solution to robust risk

Class Imbalance

The classes are very imbalanced...

Search results for “microwave”

Showing 1 - 60 of 221,489 Results

Search results for “headphone”

Showing 1 - 60 of 13,178 Results

~20x difference!

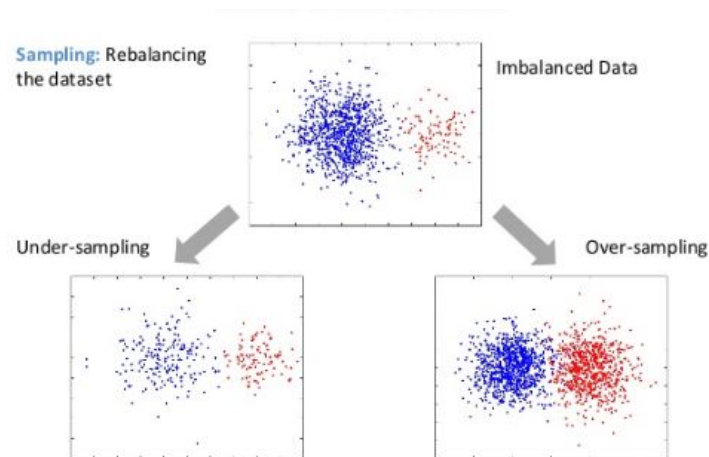
Is accuracy/risk a good measure?

Example: 99% Microwave, 1% keyboard

- Classifier A: Predicts everything as microwave
 - Accuracy: 99%
- Classifier B: Classifies all keyboards correctly, 2% error on Microwave
 - Accuracy: 98%

Previous Approaches: Data Augmentation

- SMOTE (Chawla et al. 2002)
- Under/oversampling (Zhou and Liu 2006)
- GANs (Mariani et al. 2018)



Previous Approaches: Alternative Metrics

F1 Score

$$F_1 = \left(\frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Precision: proportion of minority class predictions that are correct

Recall: proportion of true minority class samples that are predicted as minority class

Poorly understood and may not be the desired metric

Class Weighting

We formalize errors on different classes with class-conditioned risks.

$$R_y(f) = P(f(X) \neq Y \mid Y = y)$$

Class Weighting

Weighted risk is the weighted sum of the class-conditioned risks.

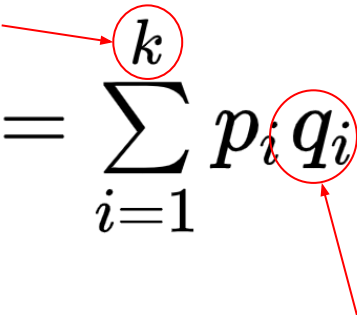
$$R_q(f) = \sum_{i=1}^k p_i q_i R_i(f)$$

$$p_i = P(Y = i)$$

$$q_i \geq 0$$

Class Weighting

However, choosing weights is a difficult task:
there are many hyperparameters to choose!

$$R_q(f) = \sum_{i=1}^k p_i q_i R_i(f)$$


Example: Credit Card Fraud

Avg cost of Mis-Classification

\$10



\$100



$$\text{Cost(fraud)} = 10 \times \text{Cost(non-fraud)}$$

Example: Credit Card Fraud

Avg cost of Mis-Classification

\$10



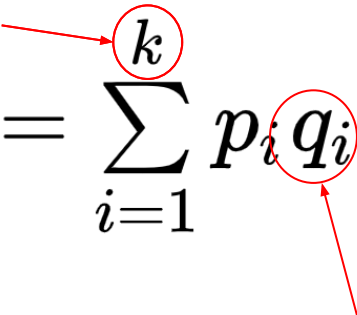
\$100



$$\text{Cost(fraud)} = 10 \times \text{Cost(non-fraud)}$$

Class Weighting

However, choosing weights is a difficult task:
there are many hyperparameters to choose!

$$R_q(f) = \sum_{i=1}^k p_i q_i R_i(f)$$


What is the effect of choosing different weightings?

- Motivation and previous approaches
- **Fundamental trade-off for different weightings**
- Formulation for robust risk on a set of weightings
- Stochastic programming solution to robust risk

Fundamental Tradeoff

Binary classification setup:

$$Y \in \{0, 1\}$$

$$\eta(x) = P(Y = 1 \mid X = x)$$

Bayes optimal classifier:

$$t_q = \frac{q_0}{q_0 + q_1}$$

$$f_q^*(x) = 1 \{ \eta(x) > t_q \}$$

Fundamental Tradeoff

Plug-in estimator:

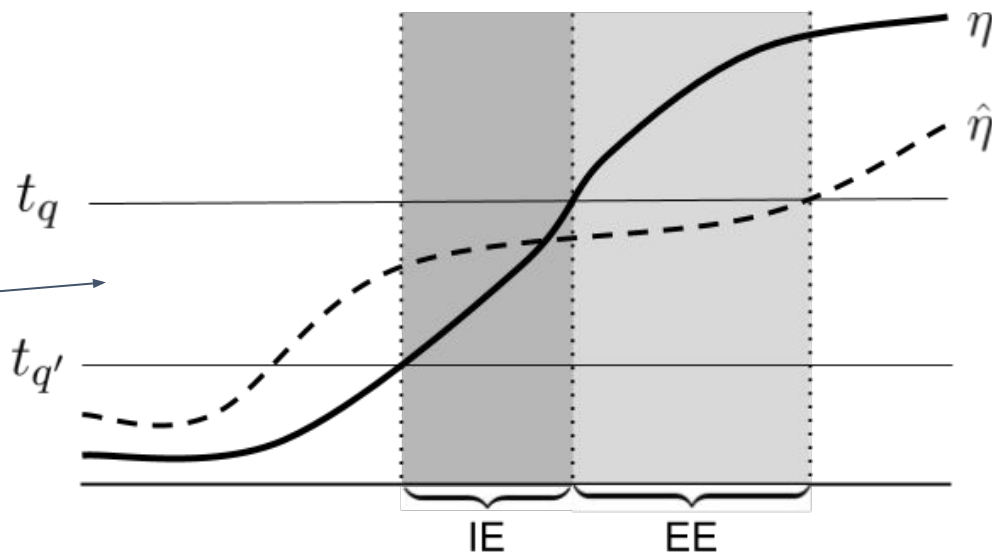
$$\hat{f}_q(x) = 1_{\{\hat{\eta}(x) > t_q\}}$$

Weighted excess risk:

$$\mathcal{E}_q(f) = R_q(f) - R_q(f_q^*)$$

Fundamental Tradeoff

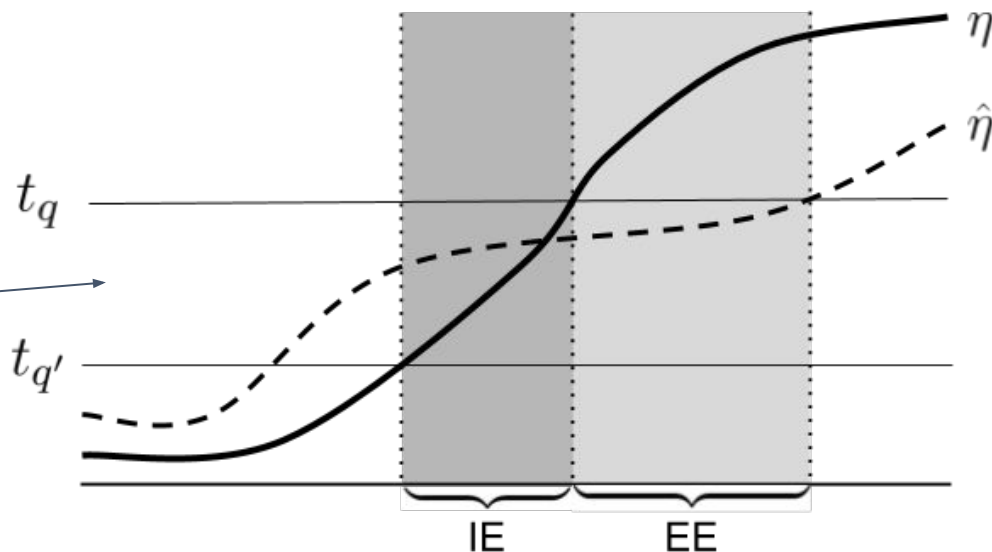
Region where
differing predictions
occur



$$\mathbb{E}\mathcal{E}_{q'}(\hat{f}_q) = \underbrace{R_{q'}(f_q^*) - R_{q'}(f_{q'}^*)}_{\text{irreducible error (IE)}} + \underbrace{\mathbb{E}R_{q'}(\hat{f}_q) - R_{q'}(f_q^*)}_{\text{estimation error (EE)}}$$

Fundamental Tradeoff

Region where
differing predictions
occur



Optimizing for one weighting inevitably reduces performance on another

$$\mathbb{E}\mathcal{E}_{q'}(\hat{f}_q) = \underbrace{R_{q'}(f_q^*) - R_{q'}(f_{q'}^*)}_{\text{irreducible error (IE)}} + \underbrace{\mathbb{E}R_{q'}(\hat{f}_q) - R_{q'}(f_q^*)}_{\text{estimation error (EE)}}$$

- Motivation and previous approaches
- Fundamental trade-off for different weightings
- **Formulation for robust risk on a set of weightings**
- Stochastic programming solution to robust risk

Robust Weighting

Define Q as a set of weightings - we define a robust risk as the maximum weighted risk over Q :

$$R_Q(f) = \max_{q \in Q} R_q(f)$$

- Motivation and previous approaches
- Fundamental trade-off for different weightings
- Formulation for robust risk on a set of weightings
- **Stochastic programming solution to robust risk**

Label CVaR

The result is label CVaR (LCVaR), a new optimization objective based on a specific robust weighted risk.

$$R_{Q_\alpha}(f) = \max_{q \in Q_\alpha} \sum_{i=1}^k p_i q_i R_i(f)$$

$$Q_\alpha = \left\{ q : \sum_{i=1}^k p_i q_i = 1, q_i \in [0, \alpha^{-1}] \right\}$$

Label CVaR

The result is label CVaR (LCVaR), a new optimization objective based on a specific robust weighted risk.

$$R_{Q_\alpha}(f) = \max_{q \in Q_\alpha} \sum_{i=1}^k p_i q_i R_i(f)$$

Each weight has a selected upper bound.

$$Q_\alpha = \left\{ q : \sum_{i=1}^k p_i q_i = 1, q_i \in [0, \alpha^{-1}] \right\}$$

Must be a probability.

LHCVaR

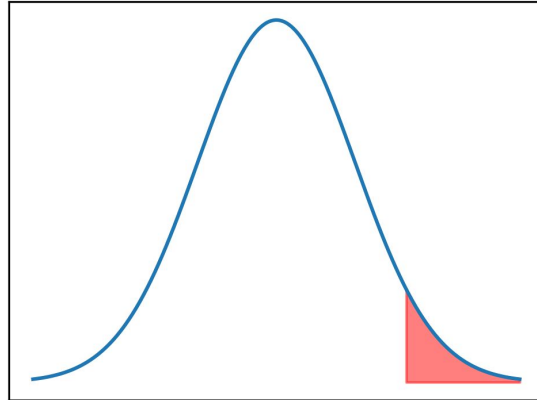
Since different classes have different sizes, we can also use different maximum weights.

$$Q_\alpha = \left\{ q : \sum_{i=1}^k p_i q_i = 1, q_i \in [0, \alpha_i^{-1}] \right\}$$

We call this version label heterogeneous CVaR (LHCVaR), since the label weights are not necessarily uniform like in LCVaR

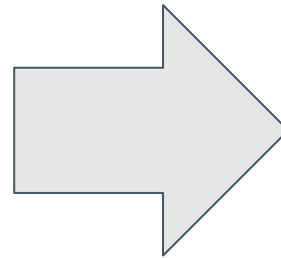
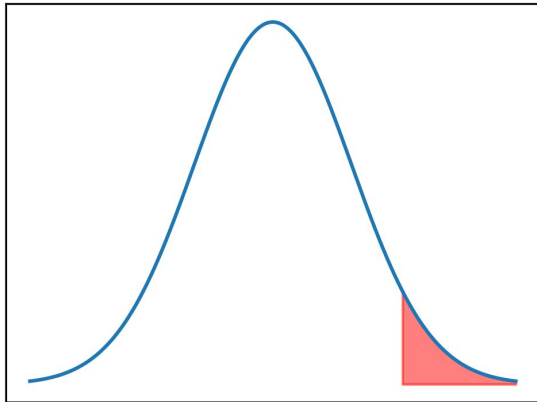
CVaR

This type of robust problem has been studied in portfolio optimization. One formulation is the α conditional value-at-risk (CVaR), which is the average loss conditional on the loss being above the $(1 - \alpha)$ -quantile.



CVaR

Main idea: instead of optimizing the worst α -proportion of losses in a portfolio, achieve good accuracy on the worst α -proportion of class labels.



Optimization

The connection to CVaR presents us with a dual form, that allows for minimization over all variables.

$$R_{Q_\alpha}(f) = \min_{\lambda \in \mathbb{R}} \mathbb{E}[\alpha_Y^{-1}(R_Y(f) - \lambda)_+ + \lambda]$$

Conclusions

- Minimizing LCVaR/LHCVaR enables good performance all weightings, rather than on a single weighting.
- LCVaR require fewer user tuned parameters.
- LCVaR/LHCVaR have dual forms that can be optimized efficiently.

Thank you!

Main equations

LCVaR:

$$R_{Q_\alpha}(f) = \max_{q \in Q_\alpha} \sum_{i=1}^k p_i q_i R_i(f)$$

$$Q_\alpha = \left\{ q : \sum_{i=1}^k p_i q_i = 1, q_i \in [0, \alpha^{-1}] \right\}$$

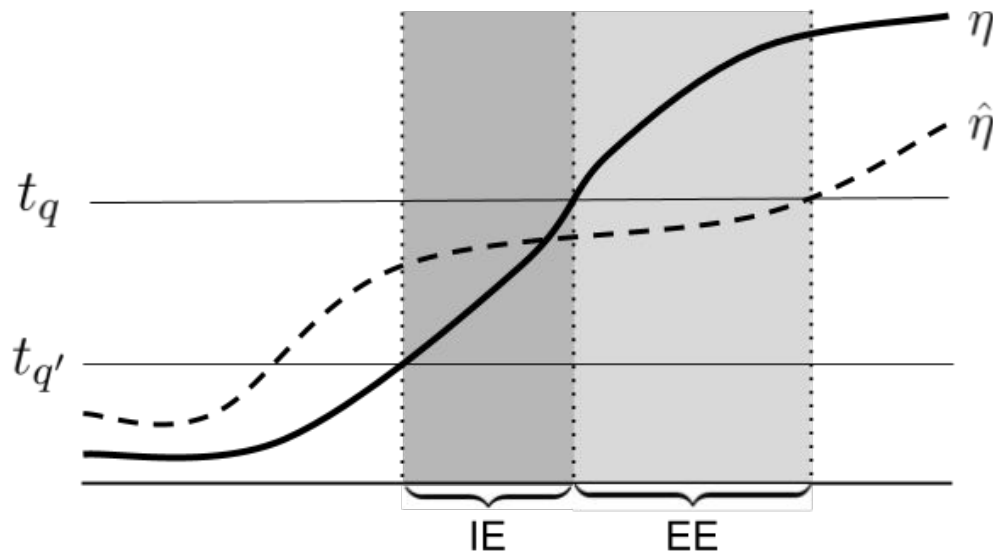
Main equations

LHCVaR:

$$R_{Q_{H_\alpha}}(f) = \max_{q \in Q_{H_\alpha}} \sum_{i=1}^k p_i q_i R_i(f)$$

$$Q_{H_\alpha} = \left\{ q : \sum_{i=1}^k p_i q_i = 1, q_i \in [0, \alpha_i^{-1}] \right\}$$

Fundamental Trade-off Summary



$$\mathbb{E}\mathcal{E}_{q'}(\hat{f}_q) = \underbrace{R_{q'}(f_q^*) - R_{q'}(f_q^*)}_{\text{irreducible error (IE)}} + \underbrace{\mathbb{E}R_{q'}(\hat{f}_q) - R_{q'}(f_q^*)}_{\text{estimation error (EE)}}$$

Hyperparameter tuning for LHCVaR

Recall that LHCVaR is the heterogeneous version of our loss
i.e. we can choose a different α for each class.

That means the number of hyperparameters scale w/ the
number of classes, which is scary.

Hyperparameter tuning for LHCVaR

It seems somewhat reasonable to choose alphas inversely proportional to the the class proportions:

$$\alpha_i = c \left(\frac{p_i^{1/\kappa}}{\sum_{i=1}^k p_i^{1/\kappa}} \right)$$

Acts as upper bound on any alpha

Temperature parameter:

As kappa goes to infinity, the alphas become closer to uniform

As kappa goes to 0 - the sharper the alphas become.

Dual form optimization tricks

Note that the dual form is non-smooth, which actually makes gradient descent a little inefficient in this case, but we can explicitly calculate λ at each step:

Dual form optimization tricks

Dual objective:

$$\min_f \min_{\lambda} \sum_{i=1}^k p_i q_i (R_i(f) - \lambda) + \lambda$$

$$\lambda = \min \left(\left\{ R_{(i)} : i \in [k], \sum_{j=1}^i p_j \alpha_j^{-1} \leq 1 \right\} \cup \{0\} \right)$$

Numerical validation

Experimental Evaluation

- Synthetic dataset, in which we simulate large class imbalance for binary classification.
- A real dataset from the UCI dataset repository, which has multiclass class imbalance.

In our experiments, we use a logistic regression model.

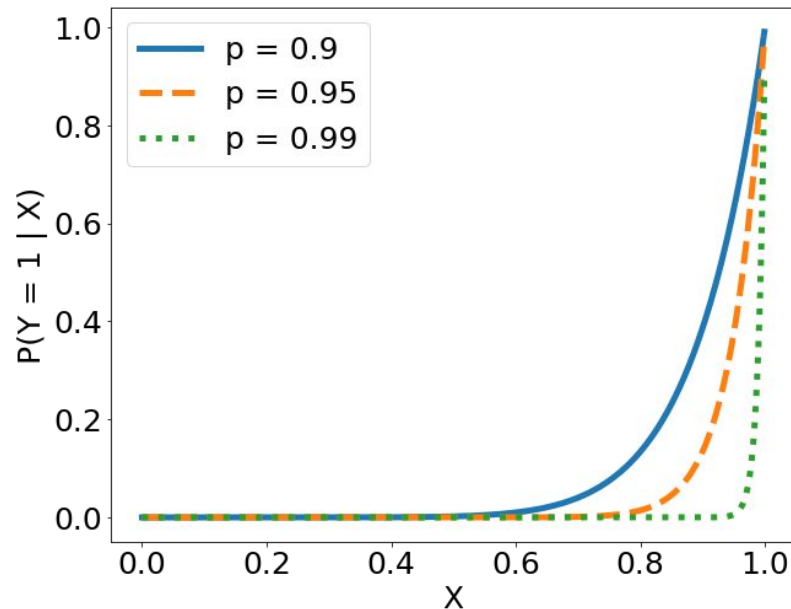
Synthetic Experiment

We generate a binary classification dataset, where we vary probability of class 0, the majority class.

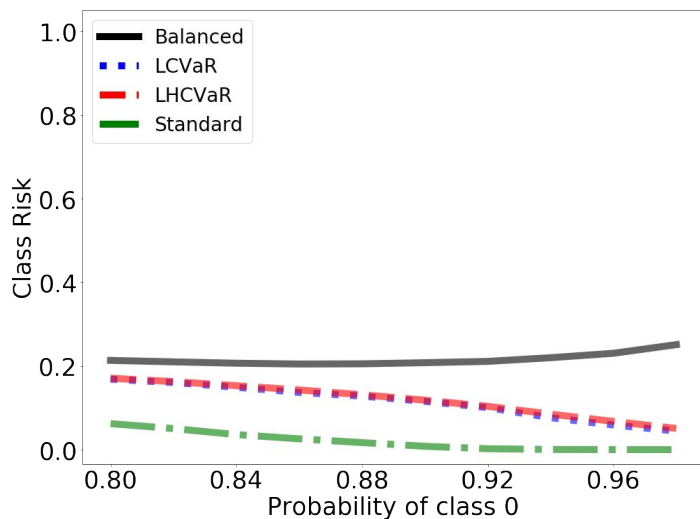
$$X \sim \text{Uniform}(0, 1)$$

$$Y \sim \text{Bernoulli}\left(X^{p/(1-p)}\right)$$

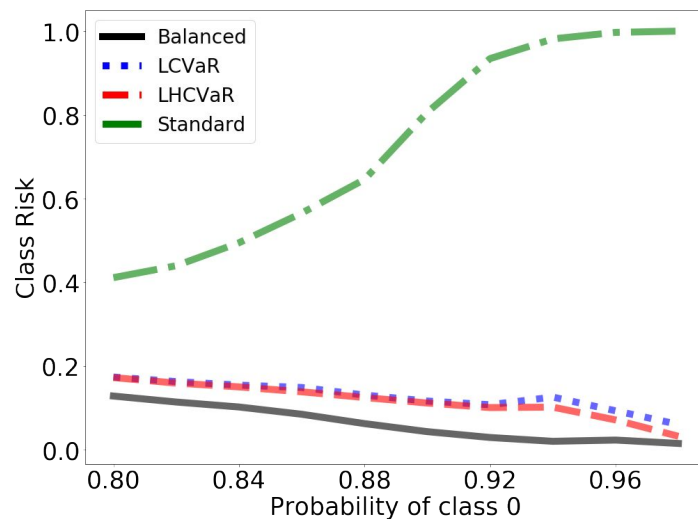
$$p = P(Y = 0)$$



Synthetic Experiment



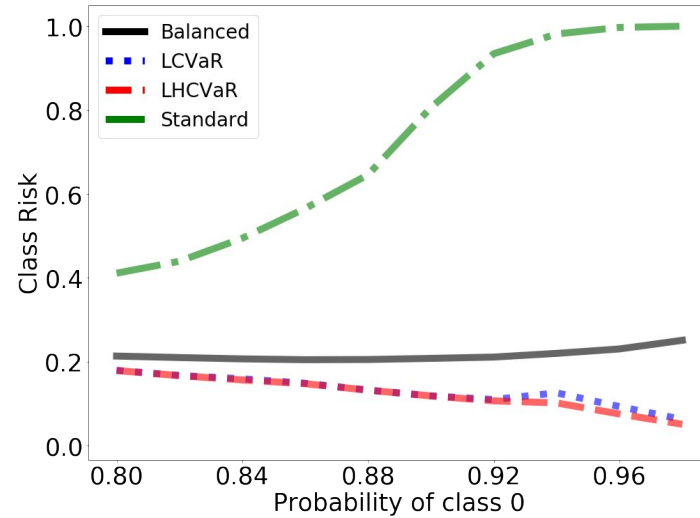
Risk on majority class



Risk on minority class

LCVaR/LHCVaR beats balanced on majority class, and standard on minority class.

Synthetic Experiment



Worst case risk

And consequently has increasingly better worst case risk as imbalance increases.

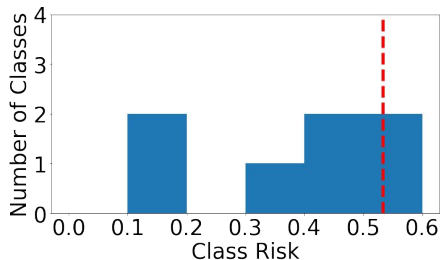
Real Data Experiment

Covertypes dataset:

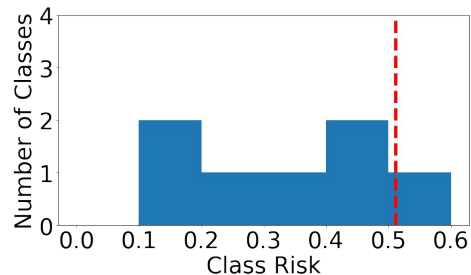
<https://archive.ics.uci.edu/ml/datasets/covertypes>

54-dimension feature set. 7 labels.

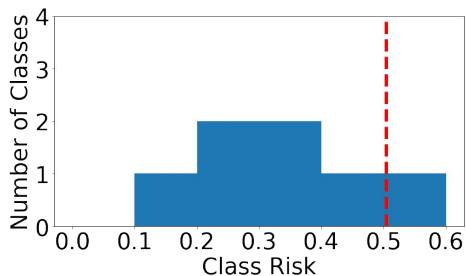
Real Data Experiment



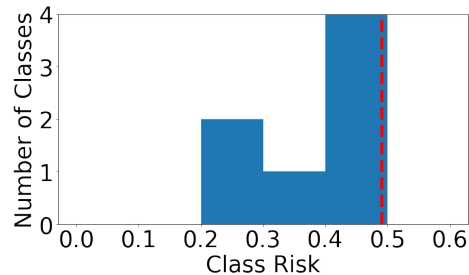
Balanced (0.5333)



Standard (0.5111)



LCVaR (0.5037)



LHCVaR (**0.4907**)

LHCVaR/LCVaR have the best worst case class risk