Classification with Rejection Based on Cost-sensitive Classification

ICML ’21

mhkang 2023-05-25

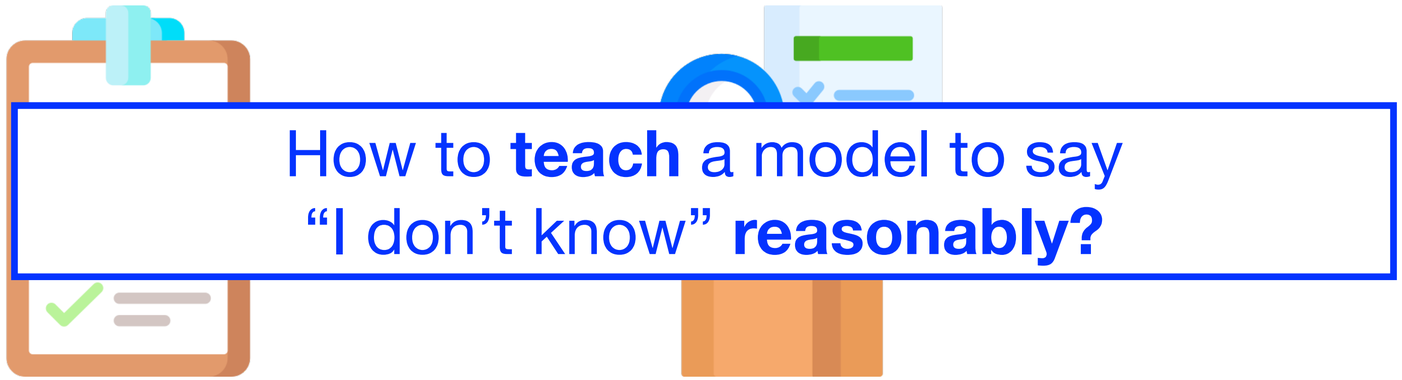
Introduction

- Mistakes in predictions can be very harmful in **error-critical applications**
  - such as medical diagnosis and product inspection

- Always answering is prune to misclassification
  - Saying **“I don’t know”** can reduce the risk of misclassification

Introduction

- Mistakes in predictions can be very harmful in **error-critical applications**
  - such as medical diagnosis and product inspection



- Always answering is prune to misclassification
  - Saying **“I don’t know”** can reduce the risk of misclassification

Binary classification

- Given: training feature-label pairs

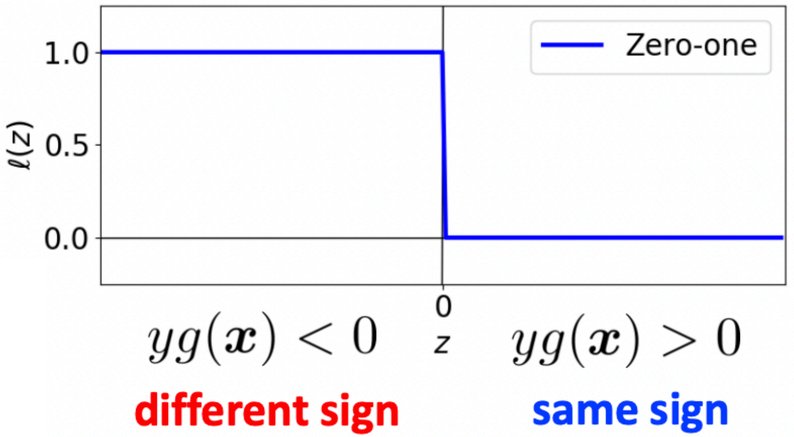
$$\{x_i, y_i\}_{i=1}^{n} \underset{i.i.d.}{\sim} p(x, y)$$

$y \in \{-1, 1\}$: label
$g : \mathbb{R}^d \to \mathbb{R}$: prediction function
$x = \mathbb{R}^d$: feature vector
$l : \mathbb{R} \to \mathbb{R}$: margin loss function
$z = yg(x)$: margin

- Goal: find g that minimizes **the expected risk**

$$R^{l_{0-1}}(g) = \mathbb{E}_{(x,y)\sim p(x,y)}[l_{0-1}(yg(x))]$$

zero-one loss



Binary classification

- Given: training feature-label pairs

$$\{x_i, y_i\}_{i=1}^{n} \underset{i.i.d.}{\sim} p(x, y)$$

$y \in \{-1, 1\}$: label
$g : \mathbb{R}^d \to \mathbb{R}$: prediction function
$x = \mathbb{R}^d$: feature vector
$l : \mathbb{R} \to \mathbb{R}$: margin loss function
$z = yg(x)$: margin

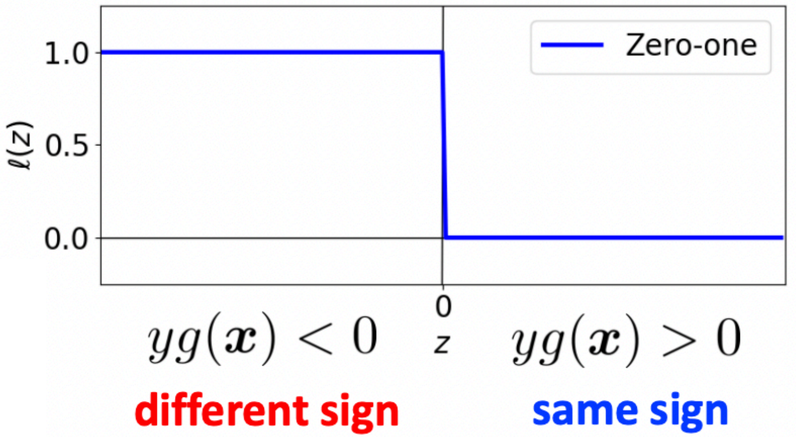
- Goal: find g that minimizes **the expected risk**

$$R^{l_{0-1}}(g) = \mathbb{E}_{(x,y)\sim p(x,y)}[l_{0-1}(yg(x))]$$

Cannot minimize the expected risk directly

No access to distribution $p(x, y)$

zero-one loss



Binary classification

- Given: training feature-label pairs

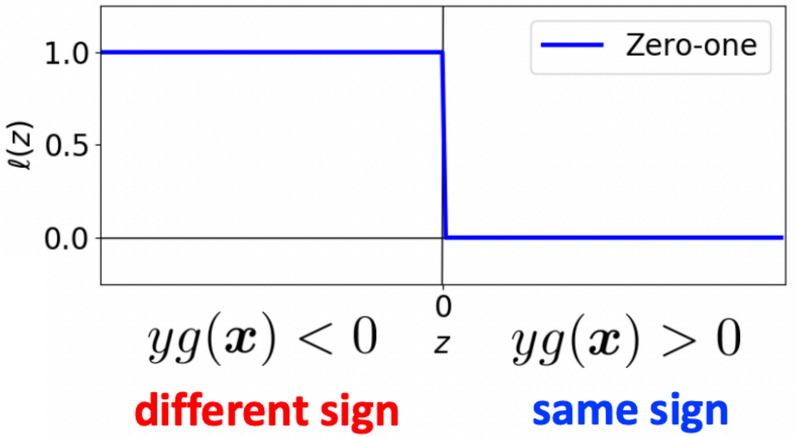
$$\{x_i, y_i\}_{i=1}^{n} \underset{i.i.d.}{\sim} p(x, y)$$

$y \in \{-1, 1\}$: label
$g : \mathbb{R}^d \to \mathbb{R}$: prediction function
$x = \mathbb{R}^d$: feature vector
$l : \mathbb{R} \to \mathbb{R}$: margin loss function
$z = yg(x)$: margin

- Goal: find g that minimizes **the empirical risk**

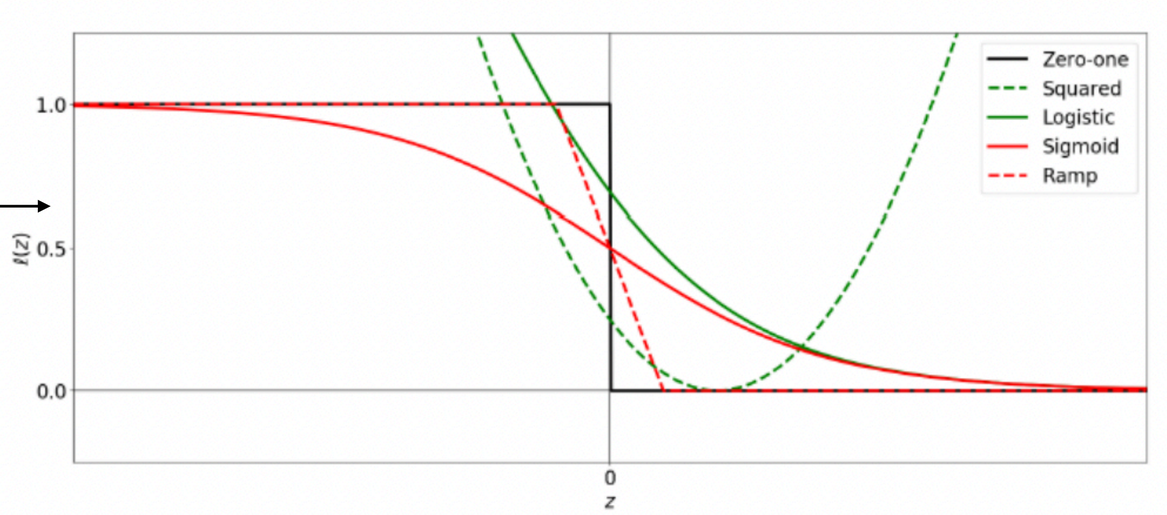
$$\hat{R}^{l_{0-1}}(g) = \frac{1}{n} \sum_{i=1}^{n} [l_{0-1}(y_i g(x_i))]$$

zero-one loss



Zero-one loss and its surrogate

$$\hat{R}^{l_{0-1}}(g) = \frac{1}{n}\sum_{i=1}^{n}[l_{0-1}(y_i g(x_i))]$$



- Minimizing $\hat{R}^{l_{0-1}}$ is NP-hard
- Surrogate losses which are easier to minimize are used in practice
  - If a loss l is classification-calibrated,
    it is ensured that minimizing $\hat{R}^{l}$ yields good g for $\hat{R}^{l_{0-1}}$

From zero-one loss to zero-one-c loss

- Define a rejection cost
  $c \in (0,0.5]$

$g : \mathbb{R}^d \rightarrow \mathbb{R}$: prediction function
$r : \mathbb{R}^d \rightarrow \{0,1\}$: rejection function

- Zero-one-c loss

$$l_{0-1-c}(y, r(x), g(x)) = \begin{cases} c & if\ r(x) = 0 \\ l_{0-1}(yg(x)) & otherwise \end{cases}$$

c < 1 → a classifier has an incentive to prefer rejection over misclassification

- Two main approaches to solve this problem
  - confidence-based approach
  - classifier-rejector approach

Confidence-based approach

- Use class-posterior $p(y \mid x)$

$$g^*(x) = p(y = 1 \mid x) - \frac{1}{2}$$

$$r^*(x) = 1_{[\max_y p(y|x) \leq (1-c)]}$$

$g : \mathbb{R}^d \to \mathbb{R}$: prediction function

$r : \mathbb{R}^d \to \{0,1\}$: rejection function

- Surrogate losses must be able to estimate $p(y \mid x)$
  - strictly stronger requirement than classification-calibration
  - can be difficult to estimate especially when using deep neural networks

Classifier-rejector approach

- Train r and g simultaneously

$\mathscr{H}$: prediction function class

$\mathscr{R}$: rejection function class

- Goal: find $(r, g) \in \mathscr{R} \times \mathscr{H}$ that minimizes

$$\hat{R}^{l_{0-1-c}}(r, g) = \frac{1}{n} \sum_{i=1}^{n} [l_{0-1-c}(y_i, r(x_i), g(x_i))]$$

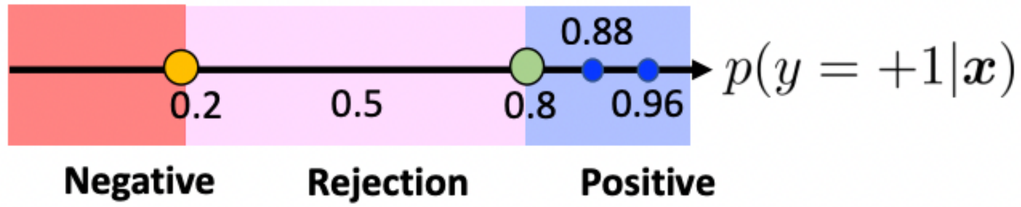
- Support a limited set of loss functions
  - In binary case, only exponential and max-hinge losses are theoretically justified
  - The multiclass extension does not work theoretically

Proposal: cost-sensitive approach

- Motivation
  - optimal decision rule for the binary classification with rejection

$$h^*(x) = \begin{cases} positive & p(y=+1\,|\,x) > 1-c \\ reject & c \le p(y=+1\,|\,x) \le 1-c \\ negative & p(y=+1\,|\,x) < c \end{cases}$$

  - we only need to know
    1. $p(y=+1\,|\,x) > 1-c$
    2. $p(y=+1\,|\,x) < c$



- Can solve it with cost sensitive classification
  - learn two cost-sensitive classifiers for $\alpha = c$ and $\alpha = 1-c$

Binary cost-sensitive classification

- Binary classification where
  - false positive cost ≠ false negative cost
    ($\alpha \in (0,1)$ = false positive cost and $1-\alpha$ = false negative cost)
  - when α=0.5, it coincides with the ordinary classification

- The optimal cost-sensitive classifier can be expressed as

$$f_{\alpha}^{*}(x) = \begin{cases} +1 & p(y=+1\,|\,x) > \alpha \\ -1 & otherwise \end{cases}$$

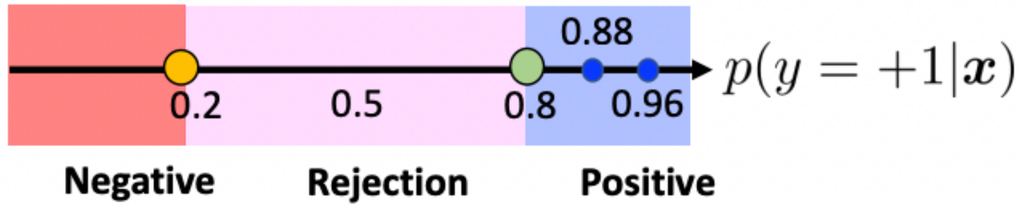
- Solving one cost-sensitive classification = knowing if $p(y=+1\,|\,x) > \alpha$

Proposal: cost-sensitive approach

- Motivation
  - optimal decision rule for the binary classification with rejection

$$h^*(x) = \begin{cases} positive & p(y=+1\,|\,x) > 1-c \\ reject & c \le p(y=+1\,|\,x) \le 1-c \\ negative & p(y=+1\,|\,x) < c \end{cases}$$

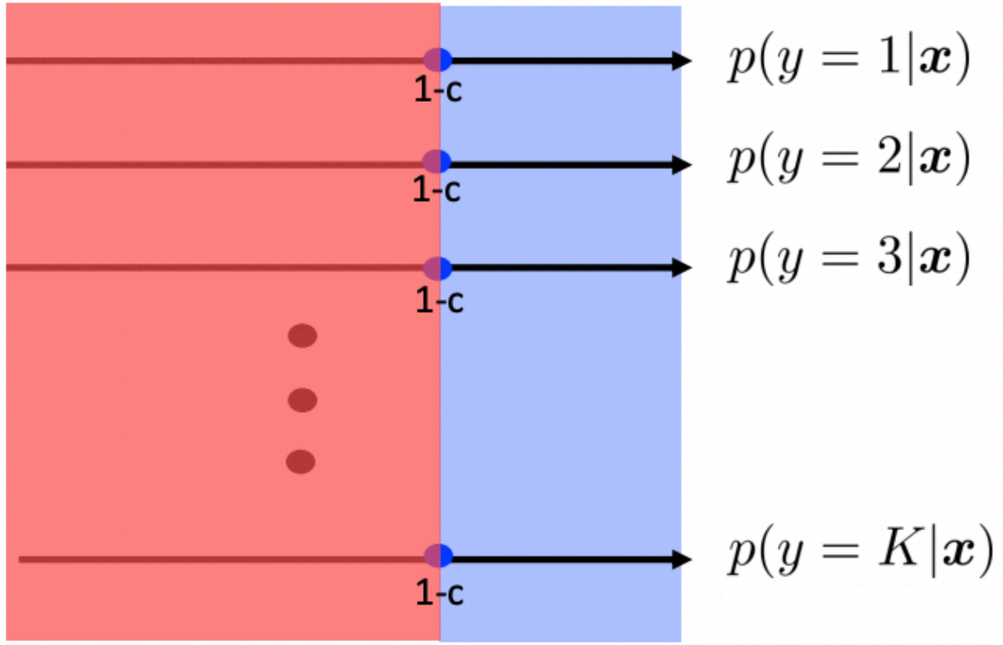
  - we only need to know
    1. $p(y=+1\,|\,x) > 1-c$
    2. $p(y=+1\,|\,x) < c$
       $(= p(y=-1\,|\,x) > 1-c)$



- Can solve classification with rejection with cost sensitive classification
  - learn two cost-sensitive classifiers for $\alpha = c$ and $\alpha = 1-c$

Extension to multi-class scenario

- Learn K one-vs-rest cost sensitive binary classifiers with $\alpha = 1 - c$

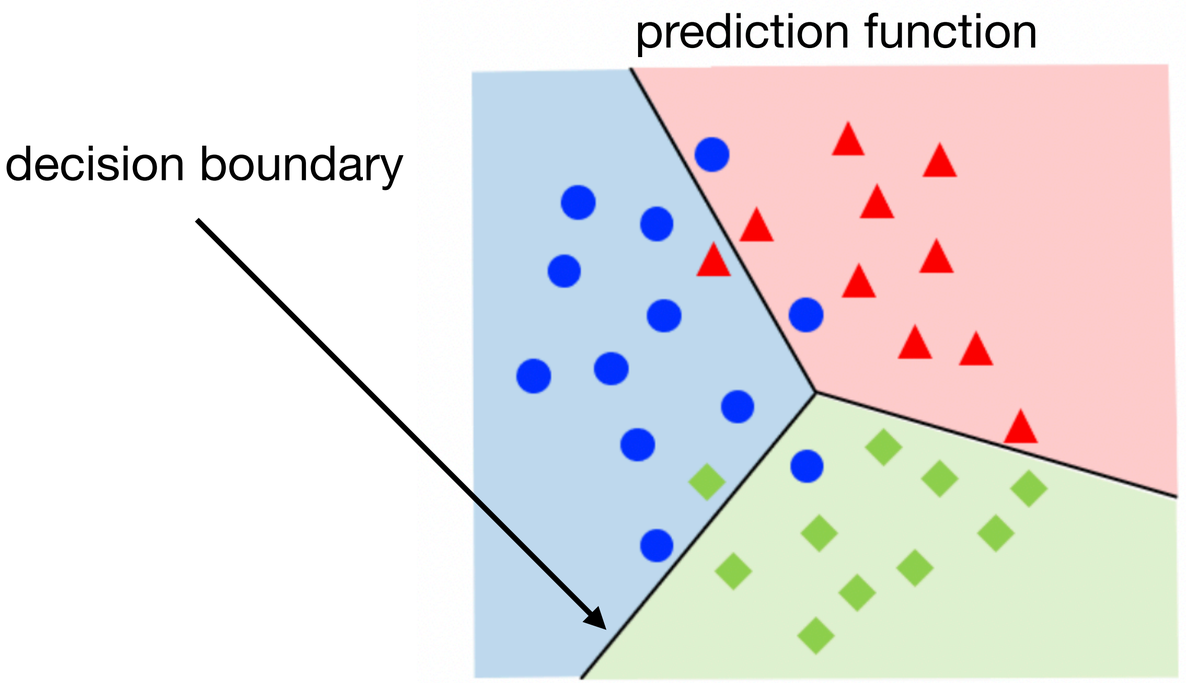


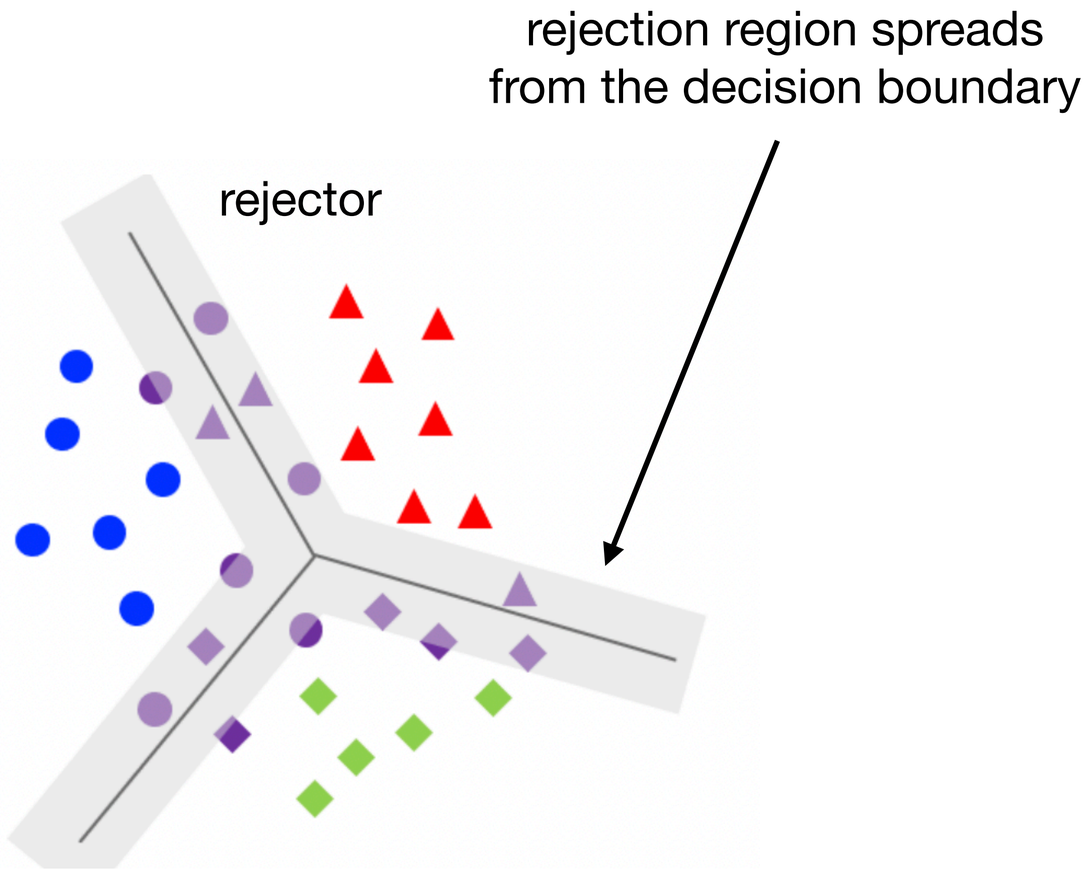
Predict if
only one classifier returns positive

Reject if
All classifiers return negative
or more than one classifier return positive

Visual interpretation

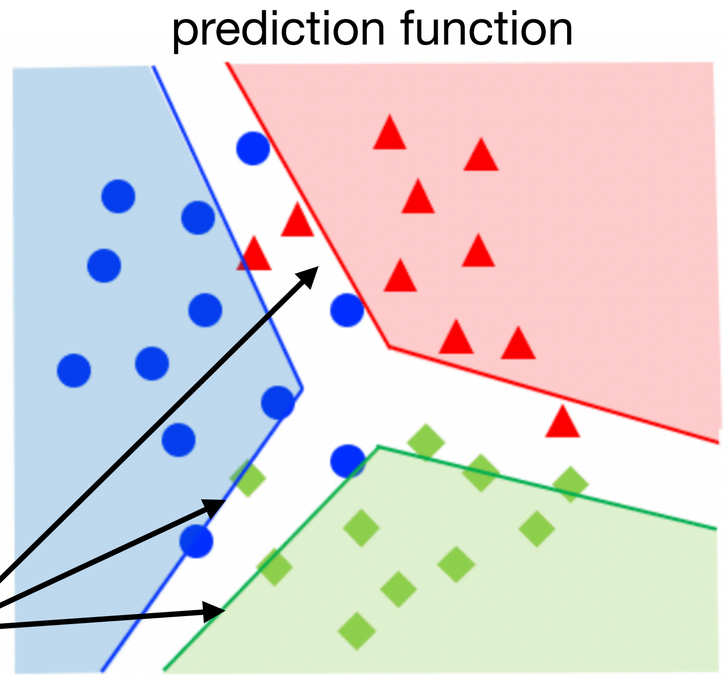
- Confidence-based approach



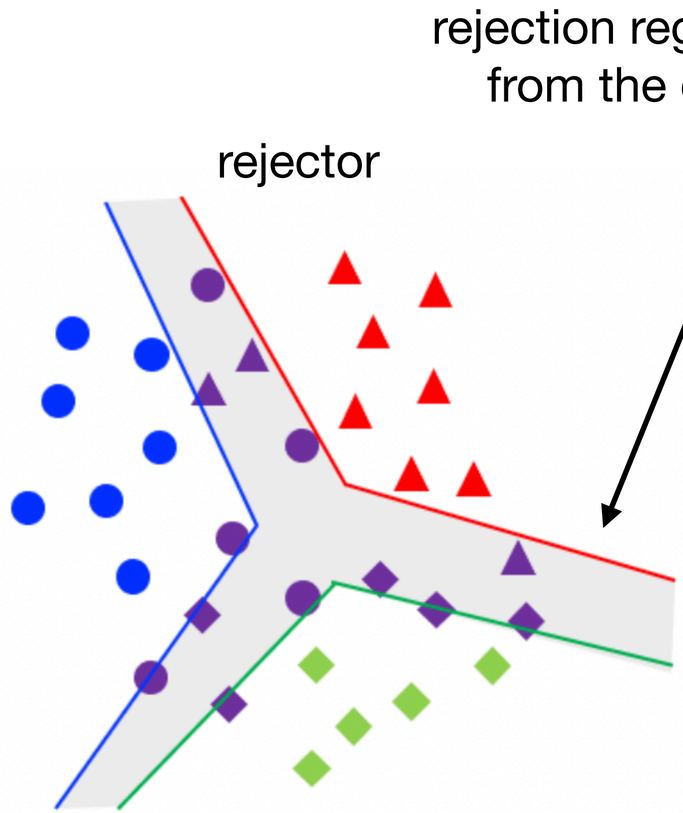


Visual interpretation

- Cost-sensitive approach



= an ensemble of cost-sensitive classifiers
for blue, red, and green classes,



Experiment setup

- Datasets
  - Subj: subjective-versus-objective classification, text dataset
  - Phishing and Spambase: tabular datasets
  - Twonorm: synthetic dataset drawn from different multivariate Gaussian distribution
  - Gisette: handwritten digit image dataset

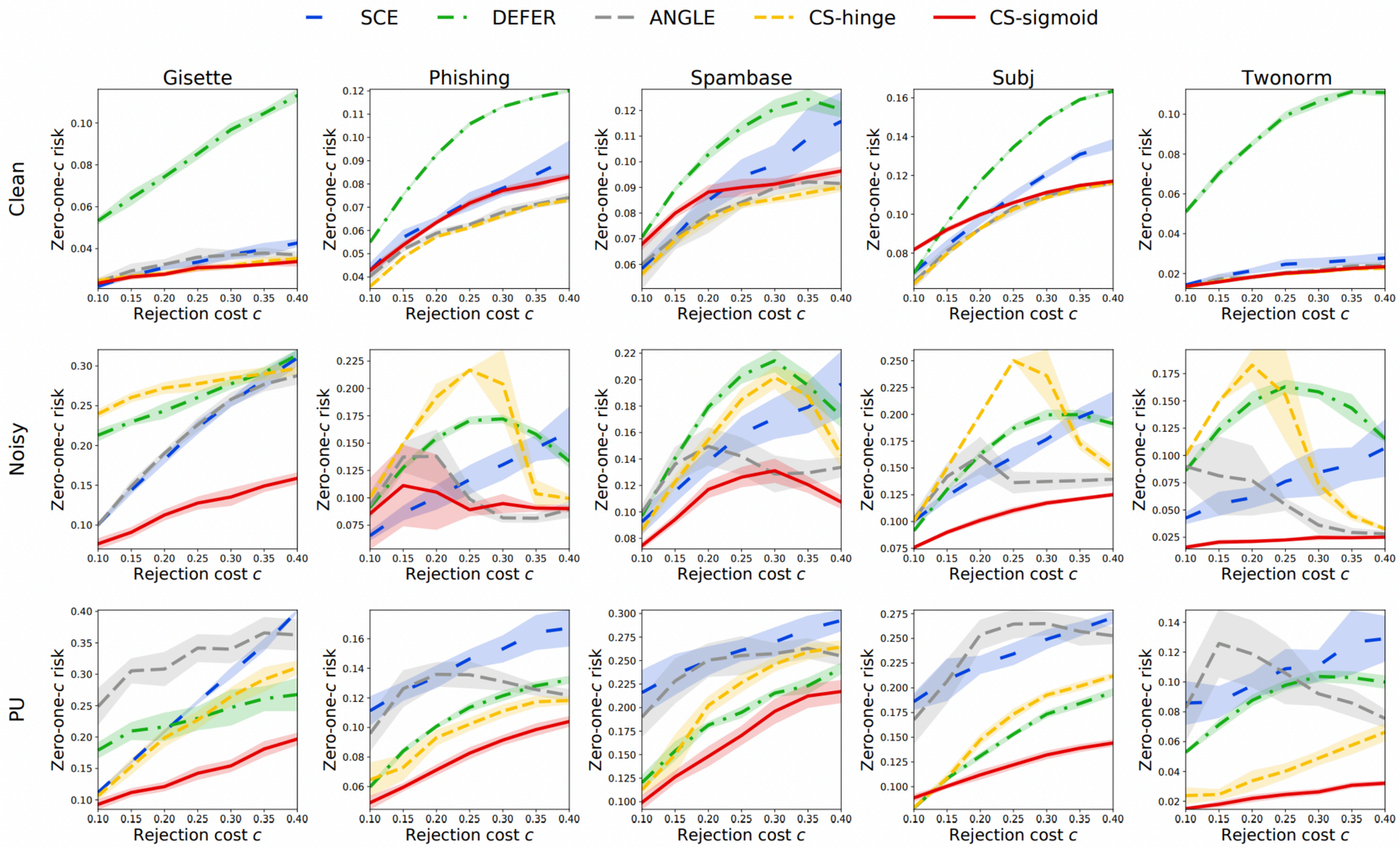
- Methods
  - CS-hinge and CS-sigmoid: cost-sensitive approach
  - SCE: softmax cross-entropy loss, confidence-based approach
  - DEFER and ANGLE: classifier-rejector approach

Experiment setup

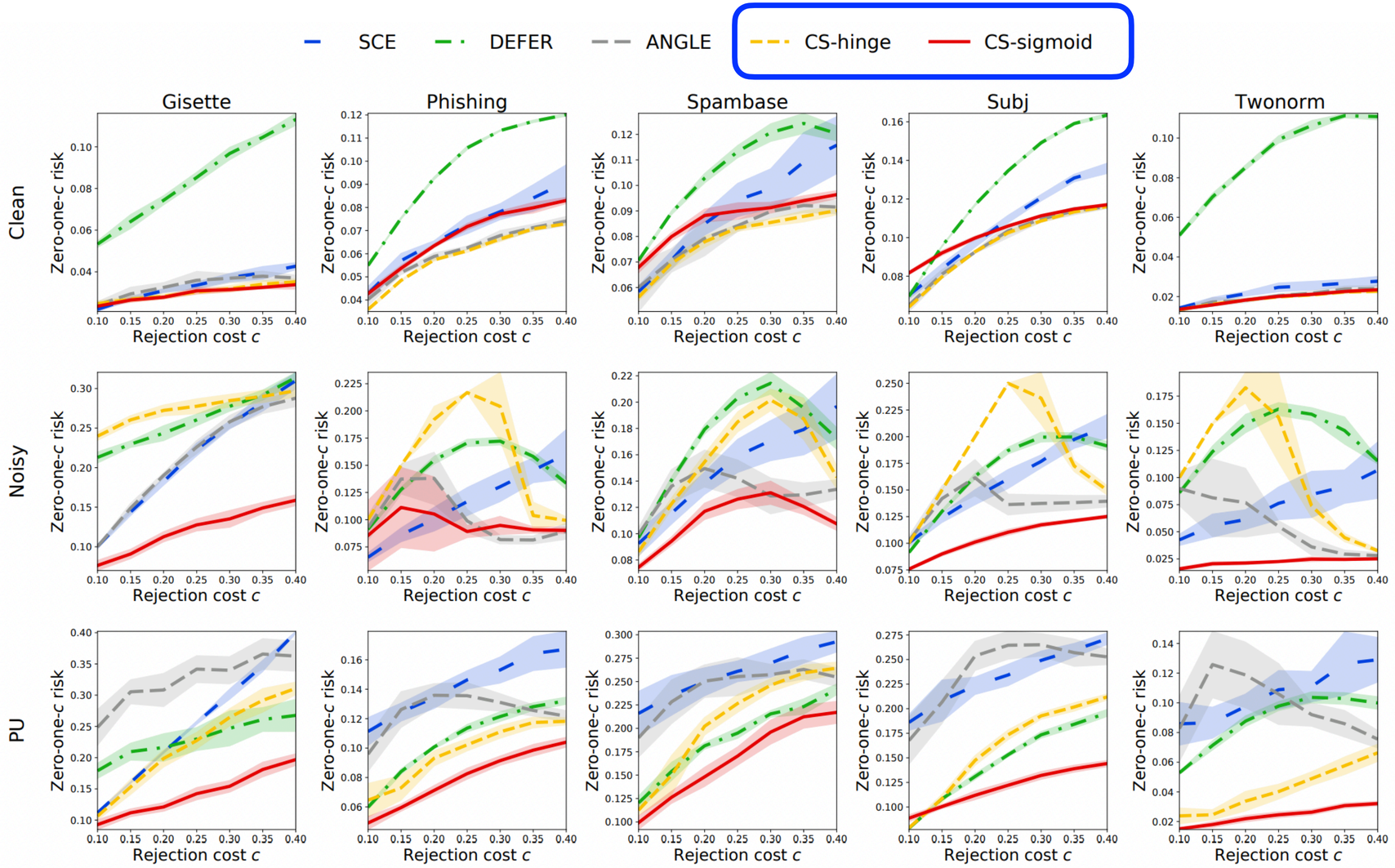
- Training Data labeling
  - clean labeled (Clean)
  - noisy labeled (Noisy)
  - positive and unlabeled (PU)

- Evaluation metric
  - empirical zero-one-c risk

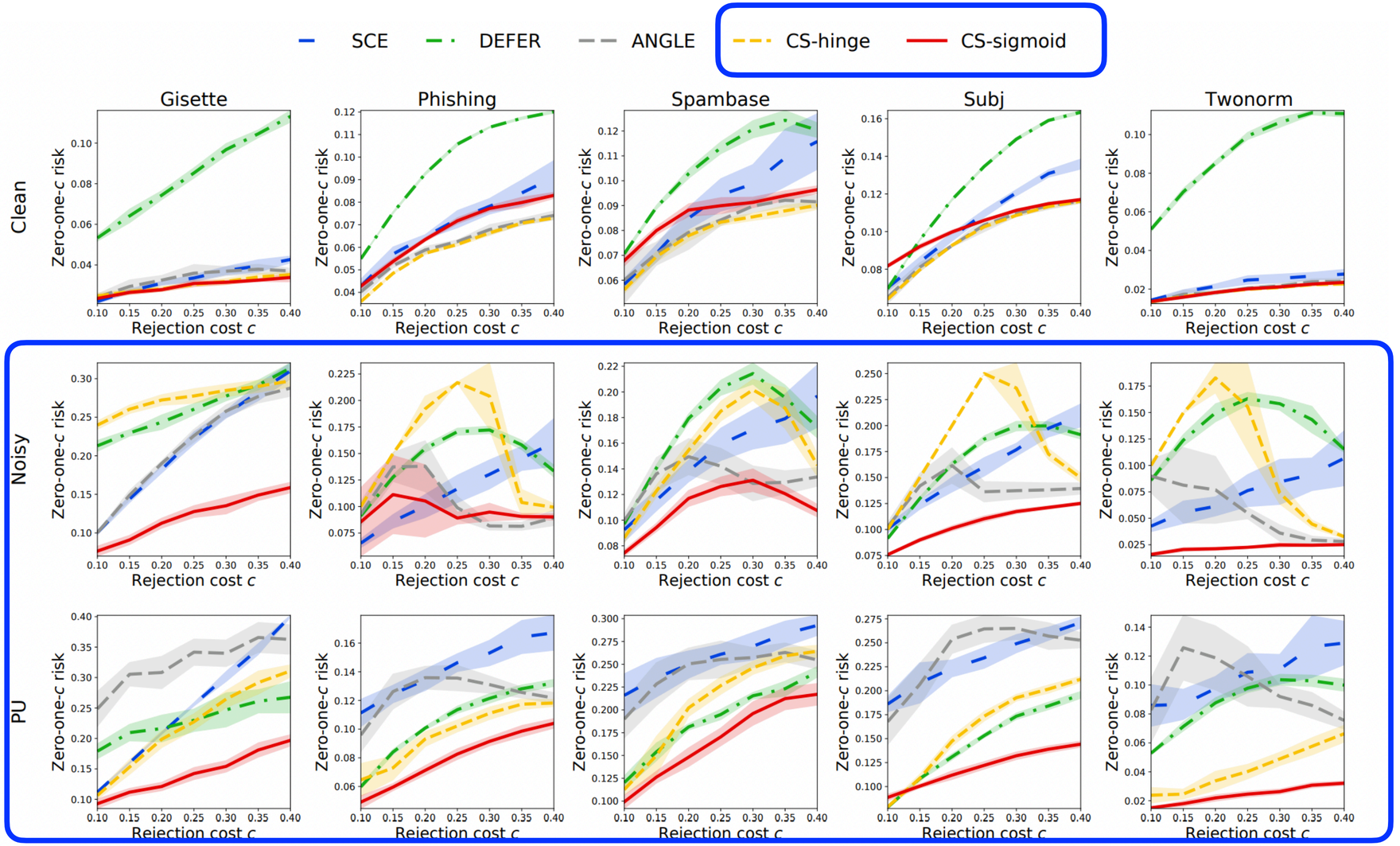
Experiment results



Experiment results



Experiment results



Conclusion

- Provide cost-sensitive approach for classification with rejection
- Show experimental results using clean-labeled, noisy-labeled, and positive and unlabeled training data
- Demonstrate the advantages of having a flexible choice of loss functions