Generalizing Consistent Multi-Class Classification with Rejection to be Compatible with Arbitrary Losses

Yuzhou Cao[1]* Tianchi Cai[2] Lei Feng[3] Lihong Gu[2] Jinjie Gu[2]
Bo An[1] Gang Niu[4] Masashi Sugiyama[4,5]

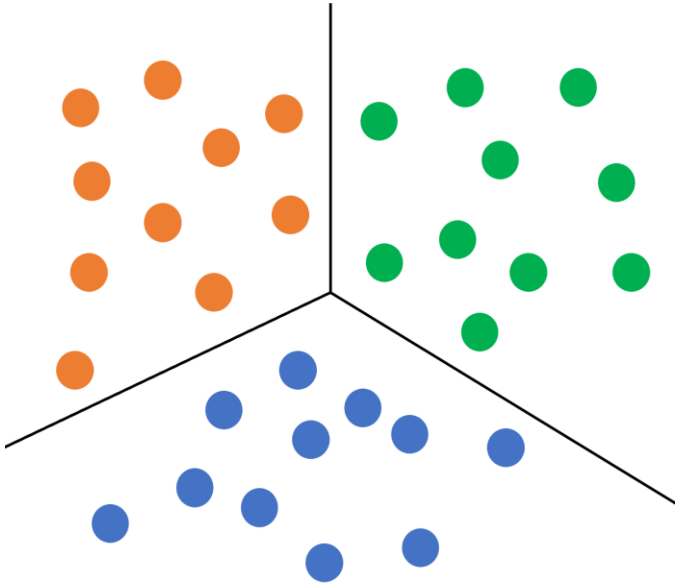
[1]Nanyang Technological University [2]Ant Group
[3]Chongqing University [4]RIKEN
[5]The University of Tokyo

*Work done when the first author was in Ant Group.

Introduction

➢ **Multi-Class Classification**

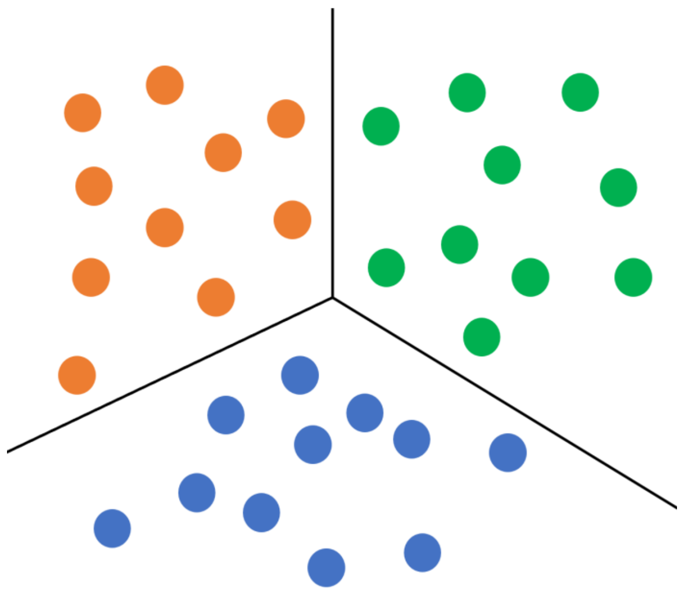
Learn a classifier to decide the exact class label of each data point.



Introduction

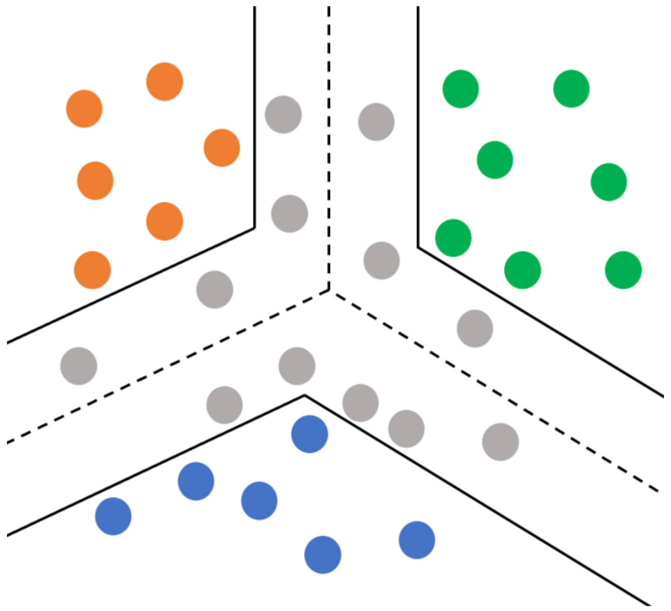
- **Multi-Class Classification**

  Learn a classifier to decide the exact class label of each data point.



- **Classification with Rejection**

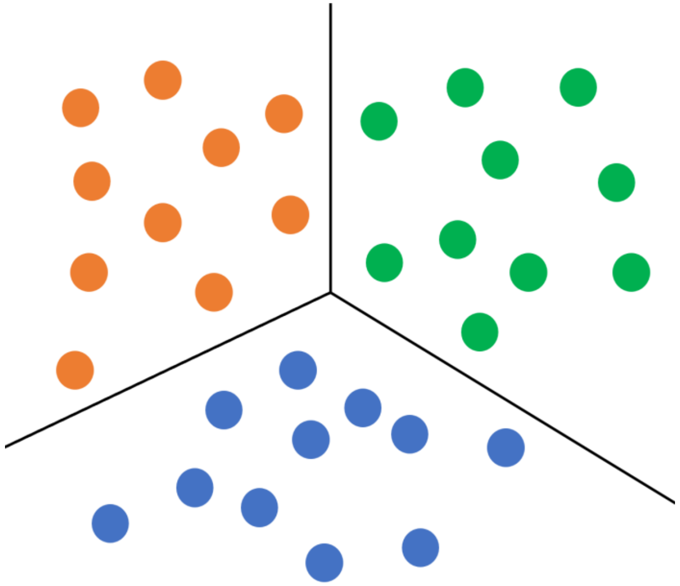
  Learn a classifier and rejector that classifies **safe** samples and rejects **risky** samples.



Introduction

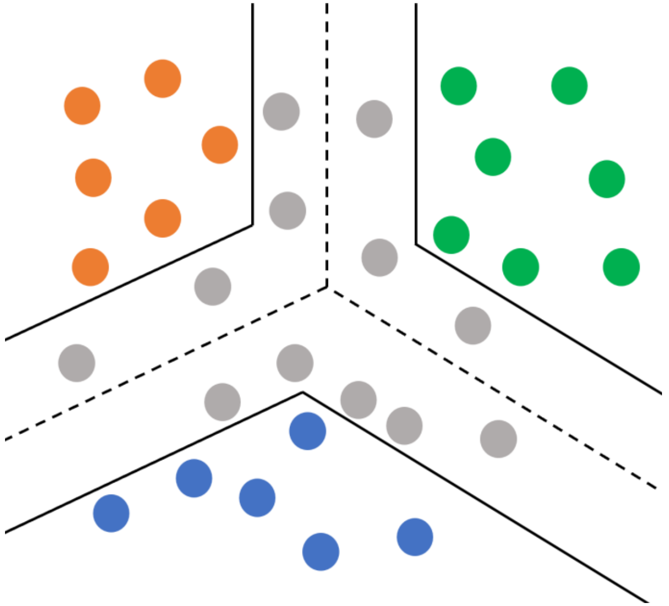
➢ Multi-Class Classification

Learn a classifier to decide the exact class label of each data point.



➢ Classification with Rejection

Learn a classifier and rejector that classifies **safe** samples and rejects **risky** samples.

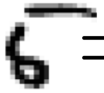
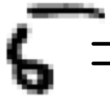


What does ‘risky’ means?

Hard Samples are Risky to be Classified

- Examples from MNIST

$p($  $=$ digit ‘7’$) = 50\%$ $\quad$ $p($  $=$ digit ‘3’$) = 50\%$

$p($  $=$ digit ‘5’$) = 50\%$ $\quad$ $p($  $=$ digit ‘6’$) = 50\%$

Minimal misclassification rate: 50%!

Optimal Rejection Rule

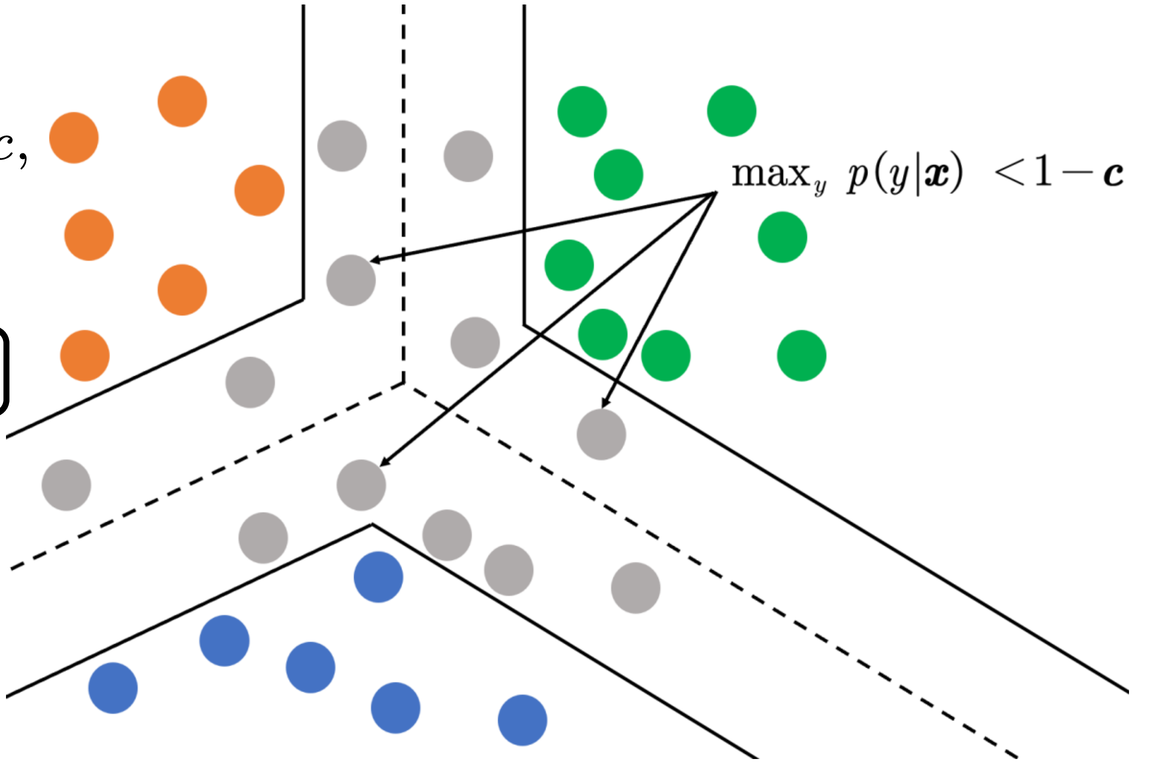
- Chow's Rule [1]

$$f^{chow}(\boldsymbol{x}) = \begin{cases} \mathrm{argmax}_y \; p(y|\boldsymbol{x}), & \text{if } \max_y p(y|\boldsymbol{x}) > 1 - c, \\ \text{reject}, & \text{else.} \end{cases}$$

Optimal Rejection Rule

- **Chow's Rule [1]**

$$f^{chow}(\boldsymbol{x}) = \begin{cases} \mathrm{argmax}_y \ p(y|\boldsymbol{x}), & \text{if } \max_y p(y|\boldsymbol{x}) > 1 - c, \\ \text{reject}, & \text{else.} \end{cases}$$

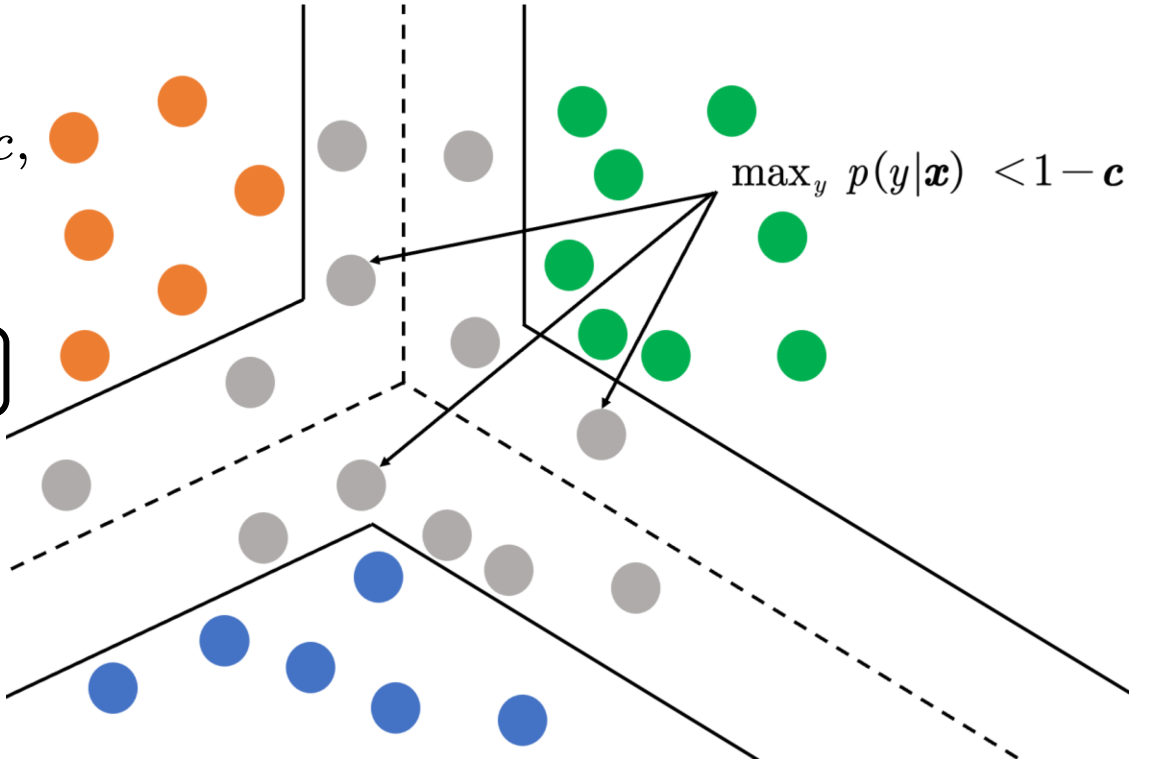


Rejecting samples with misclassification rate > c

Optimal Rejection Rule

➢ Chow's Rule [1]

$$f^{chow}(\boldsymbol{x}) = \begin{cases} \mathrm{argmax}_y \ p(y|\boldsymbol{x}), & \text{if } \max_y p(y|\boldsymbol{x}) > 1-c, \\ \text{reject}, & \text{else.} \end{cases}$$



Rejecting samples with misclassification rate > c

➢ Risk Minimization Framework:

$$R_{01c}(f) = \mathbb{E}_{p(\boldsymbol{x},y)}[\ell_{01c}(f(\boldsymbol{x}),y)]$$

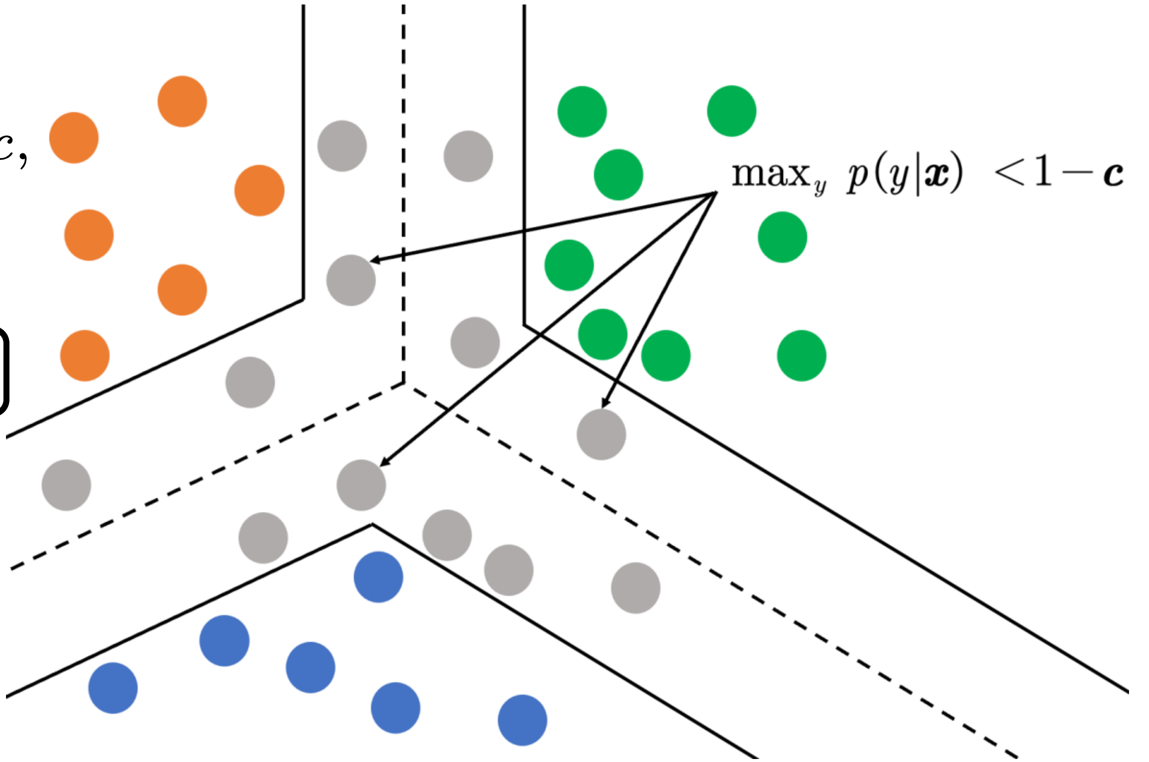
where $\ell_{01c}(f(\boldsymbol{x}),y) = \begin{cases} c, & f(\boldsymbol{x}) = \text{reject}, \\ \mathbb{I}(f(\boldsymbol{x}) \neq y), & \text{else.} \end{cases}$

Optimal Rejection Rule

- **Chow's Rule [1]**

$$f^{chow}(\boldsymbol{x}) = \begin{cases} \operatorname{argmax}_y \; p(y|\boldsymbol{x}), & \text{if } \max_y p(y|\boldsymbol{x}) > 1-c, \\ \text{reject}, & \text{else.} \end{cases}$$

Rejecting samples with misclassification rate > c



- **Risk Minimization Framework:**

$$R_{01c}(f) = \mathbb{E}_{p(\boldsymbol{x},y)}[\ell_{01c}(f(\boldsymbol{x}),y)]$$

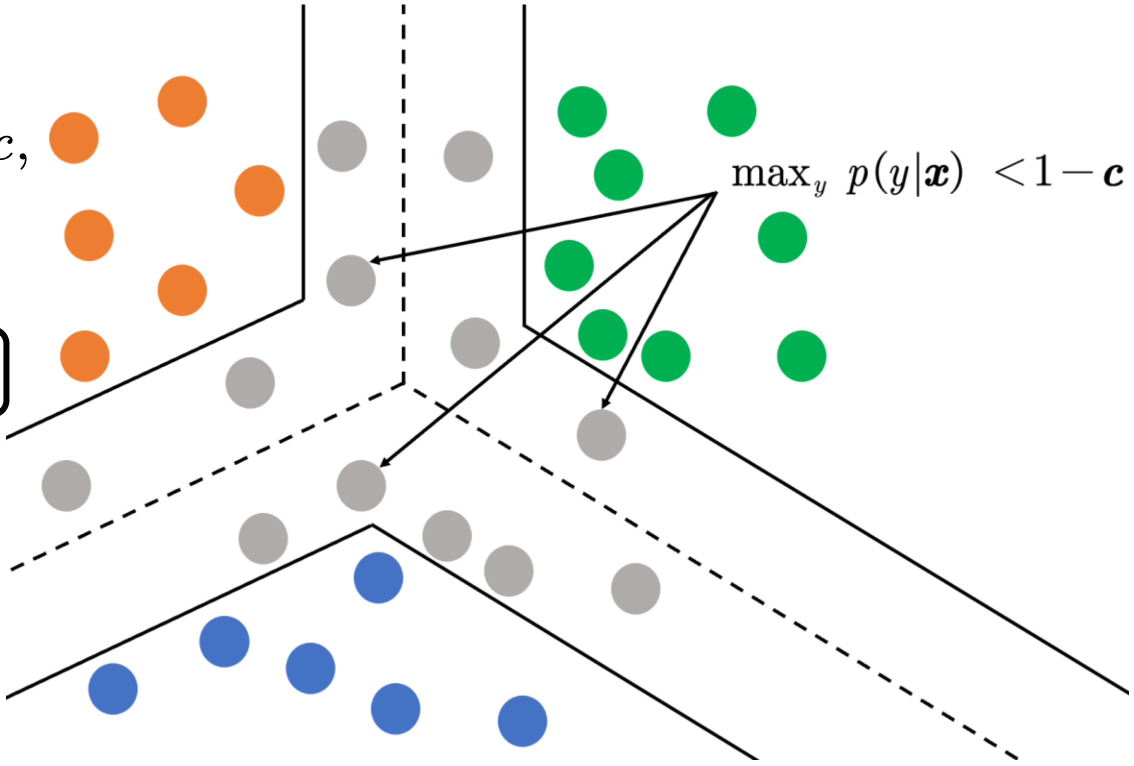
where $\ell_{01c}(f(\boldsymbol{x}),y) = \begin{cases} c, & f(\boldsymbol{x}) = \text{reject}, \\ \mathbb{I}(f(\boldsymbol{x}) \neq y), & \text{else.} \end{cases}$

$$\boxed{\operatorname{argmin}_f R_{01c}(f) = f^{chow}}$$

Optimal Rejection Rule

➢ Chow's Rule [1]

$$f^{chow}(\boldsymbol{x}) = \begin{cases} \mathrm{argmax}_y \; p(y|\boldsymbol{x}), & \text{if } \max_y p(y|\boldsymbol{x}) > 1 - c, \\ \text{reject}, & \text{else.} \end{cases}$$



Rejecting samples with misclassification rate > c

➢Risk Minimization Framework:

$$R_{01c}(f) = \mathbb{E}_{p(\boldsymbol{x},y)}[\ell_{01c}(f(\boldsymbol{x}), y)]$$

where $\ell_{01c}(f(\boldsymbol{x}), y) = \begin{cases} c, & f(\boldsymbol{x}) = \text{reject}, \\ \mathbb{I}(f(\boldsymbol{x}) \neq y), & \text{else.} \end{cases}$

$$\boxed{\mathrm{argmin}_f R_{01c}(f) = f^{chow}}$$

Intractable! (NP-hard)

Calibrated Surrogate Losses

- Class-Posterior Probability Estimation Based [2]:

  $\mathrm{argmax}_y \ \hat{p}(y|\boldsymbol{x}) \rightarrow 1$ Severe overconfidence!

Calibrated Surrogate Losses

- Class-Posterior Probability Estimation Based [2]:

  $\mathrm{argmax}_y\ \hat{p}(y|\boldsymbol{x}) \rightarrow 1$ Severe overconfidence!

- Cost-Sensitive Learning Based [3], Learning to Defer [4]:

  Restriction on loss functions (OvA Loss, CE Loss)

Problem Reduction: K+1 Class Classification

- A New Data Generation Distribution:

$$\tilde{p}(\boldsymbol{x}, \tilde{y}) = \begin{cases} \dfrac{p(\boldsymbol{x}, y)}{2-c}, & \tilde{y} \in \{1, 2, \ldots, K\}, \\ \dfrac{(1-c)\, p(\boldsymbol{x})}{2-c}, & \tilde{y} = \text{rejected}. \end{cases}$$

Problem Reduction: K+1 Class Classification

- A New Data Generation Distribution:

$$\tilde{p}(\boldsymbol{x},\tilde{y}) = \begin{cases} \dfrac{p(\boldsymbol{x},y)}{2-c}, & \tilde{y} \in \{1,2,\ldots,K\}, \\ \dfrac{(1-c)p(\boldsymbol{x})}{2-c}, & \tilde{y} = \text{rejected}. \end{cases}$$

$$\tilde{R}_{01}(f) = \mathbb{E}_{\tilde{p}(\boldsymbol{x},\tilde{y})}[\mathbb{I}(f(\boldsymbol{x}) \neq \tilde{y})]$$

$$\boxed{\operatorname{argmin}_f \tilde{R}_{01}(f) = f^{chow}}$$

Problem Reduction: K+1 Class Classification

- A New Data Generation Distribution:

$$\tilde{p}(\boldsymbol{x},\tilde{y}) = \begin{cases} \dfrac{p(\boldsymbol{x},y)}{2-c}, & \tilde{y} \in \{1,2,\dots,K\}, \\ \dfrac{(1-c)p(\boldsymbol{x})}{2-c}, & \tilde{y} = \text{rejected}. \end{cases}$$

$$\tilde{R}_{01}(f) = \mathbb{E}_{\tilde{p}(\boldsymbol{x},\tilde{y})}\left[\mathbb{I}(f(\boldsymbol{x}) \neq \tilde{y})\right]$$

$$\boxed{\operatorname{argmin}_f \tilde{R}_{01}(f) = f^{chow}}$$

$$\boxed{R_{01c}(f) - R_{01c}^{*} = (2-c)\left(\tilde{R}_{01}(f) - \tilde{R}_{01}^{*}\right)}$$

Problem Reduction: K+1 Class Classification

$$\operatorname{argmin}_f \tilde{R}_{01}(f) = f^{chow}$$

$$R_{01c}(f) - R^*_{01c} = (2-c)\left(\tilde{R}_{01}(f) - \tilde{R}^*_{01}\right)$$

K+1 class classification

$$\min_f \tilde{R}_{01}(f)$$

Equivalence!

K class classification with rejection

$$\min_f R_{01c}(f)$$

Problem Reduction: K+1 Class Classification

- **How to $\min_f \tilde{R}_{01}(f)$? Not trivial!**

  ①. No data from the class 'rejected'!

  ②. Optimization of 0-1 loss is NP-hard!

Problem Reduction: K+1 Class Classification

- **How to $\min_f \tilde{R}_{01}(f)$? Not trivial!**

  ①. No data from the class 'rejected'!

  ②. Optimization of 0-1 loss is NP-hard!

- $\min_{g} R_{L_c^\Phi}(\boldsymbol{g}) = \mathbb{E}_{p(\boldsymbol{x},y)}[L_\Phi(\boldsymbol{g}(\boldsymbol{x}),y)]$ **instead!**

  where $L_c^\Phi(\boldsymbol{u},y) = \Phi(\boldsymbol{u},y) + (1-c)\Phi(\boldsymbol{u},K+1)$ $\quad f(\boldsymbol{x}) = \begin{cases} \text{reject}, & \operatorname{argmax}_y \boldsymbol{u}_y = K+1 \\ \operatorname{argmax}_y \boldsymbol{u}_y, & \text{else.} \end{cases}$

- **Motivation:** $R_{L_c^\Phi}(\boldsymbol{g}) = (2-c)\tilde{R}_\Phi(\boldsymbol{g})$

  where $\tilde{R}_\Phi(\boldsymbol{g}) = \mathbb{E}_{\tilde{p}(\boldsymbol{x},\tilde{y})}[\Phi(\boldsymbol{g}(\boldsymbol{x}),\tilde{y})]$

Problem Reduction: K+1 Class Classification

- **How to $\min_f \tilde{R}_{01}(f)$? Not trivial!**

  ①. No data from the class 'rejected'!

  ②. Optimization of 0-1 loss is NP-hard!

- $\min_{g} R_{L_c^\Phi}(\boldsymbol{g}) = \mathbb{E}_{p(\boldsymbol{x},y)}[L_\Phi(\boldsymbol{g}(\boldsymbol{x}),y)]$ **instead!**

  where $L_c^\Phi(\boldsymbol{u},y) = \Phi(\boldsymbol{u},y) + (1-c)\Phi(\boldsymbol{u},K+1)$ $\quad f(\boldsymbol{x}) = \begin{cases} \text{reject}, & \operatorname{argmax}_y \boldsymbol{u}_y = K+1 \\ \operatorname{argmax}_y \boldsymbol{u}_y, & \text{else.} \end{cases}$

- **Motivation:** $R_{L_c^\Phi}(\boldsymbol{g}) = (2-c)\tilde{R}_\Phi(\boldsymbol{g})$

  where $\tilde{R}_\Phi(\boldsymbol{g}) = \mathbb{E}_{\tilde{p}(\boldsymbol{x},\tilde{y})}[\Phi(\boldsymbol{g}(\boldsymbol{x}),\tilde{y})]$

Main Result:
Any classification-calibrated surrogate Φ can make L_c^{Φ} calibrated w.r.t. ℓ_{01c}.

$$R_{L^\Phi_c}(\boldsymbol{g}_i) \to R^*_{L^\Phi_c} \Rightarrow R_{01c}(\operatorname{argmax}_i \boldsymbol{g}_i) \to R^*_{01c}$$

More Discoveries

- Regret Transfer Bound,
- Estimation Error Bound,
- Analysis of GCE Loss,
- Instance-Dependent Cost
- Experimental Results...

References

[1]. C. Chow. On optimum recognition error and reject tradeoff. IEEE Transactions on Information Theory, 16(1):41–46, 1970.

[2]. Chenri Ni, Nontawat Charoenphakdee, Junya Honda, and Masashi Sugiyama. On the calibration of multiclass classification with rejection. In NeurIPS, 2019.

[3]. Nontawat Charoenphakdee, Zhenghang Cui, Yivan Zhang, and Masashi Sugiyama. Classification with rejection based on cost-sensitive classification. In ICML, 2021.

[4]. Hussein Mozannar and David A. Sontag. Consistent estimators for learning to defer to an expert. In ICML, 2020.