MLSS2024@Okinawa, Japan March 8, 2024

Machine Learning from Weak Supervision:

An Empirical Risk Minimization Approach

Masashi Sugiyama

RIKEN Center for Advanced Intelligence Project/
The University of Tokyo, Japan

https://www.ms.k.u-tokyo.ac.jp/sugi/

RIKEN AIP

東京大学 THE UNIVERSITY OF TOKYO

What Is This Lecture about?

- Machine learning from big labeled data has been highly successful.
  - Speech recognition, image understanding, natural language translation, recommendation, …

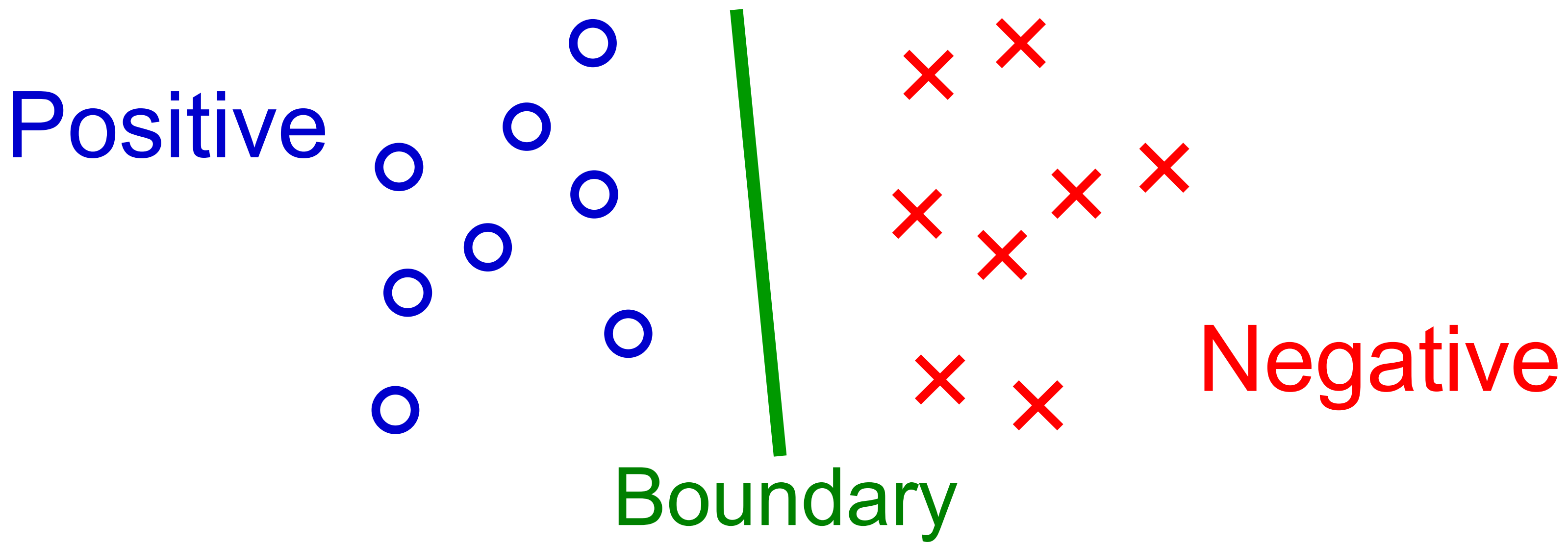
- However, there are various applications where massive labeled data is not available.
  - Medicine, disaster, robots, brain, …

What Is This Lecture about?

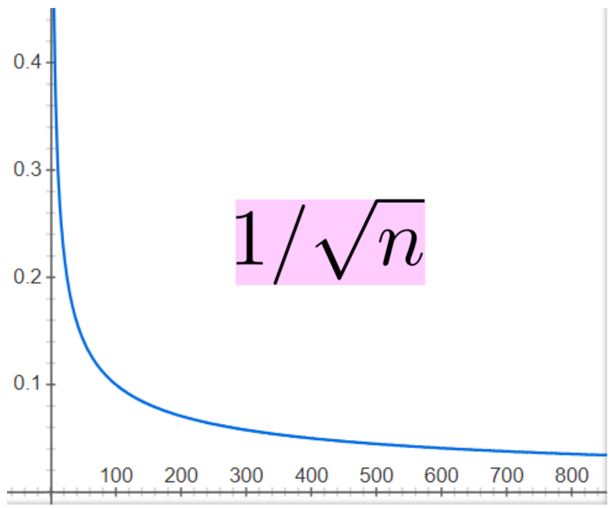
- There are many approaches to coping with the label-cost problem:
  - Improve data collection (e.g., crowdsourcing)
  - Use a simulator to generate pseudo data
  - Use domain knowledge (i.e., engineering)
  - Use cheap but weak data (e.g., unlabeled)

- Disclaimer:
  - There are many great works on weakly supervised learning.
  - Coverage of this lecture is biased and limited.

Binary Supervised Classification



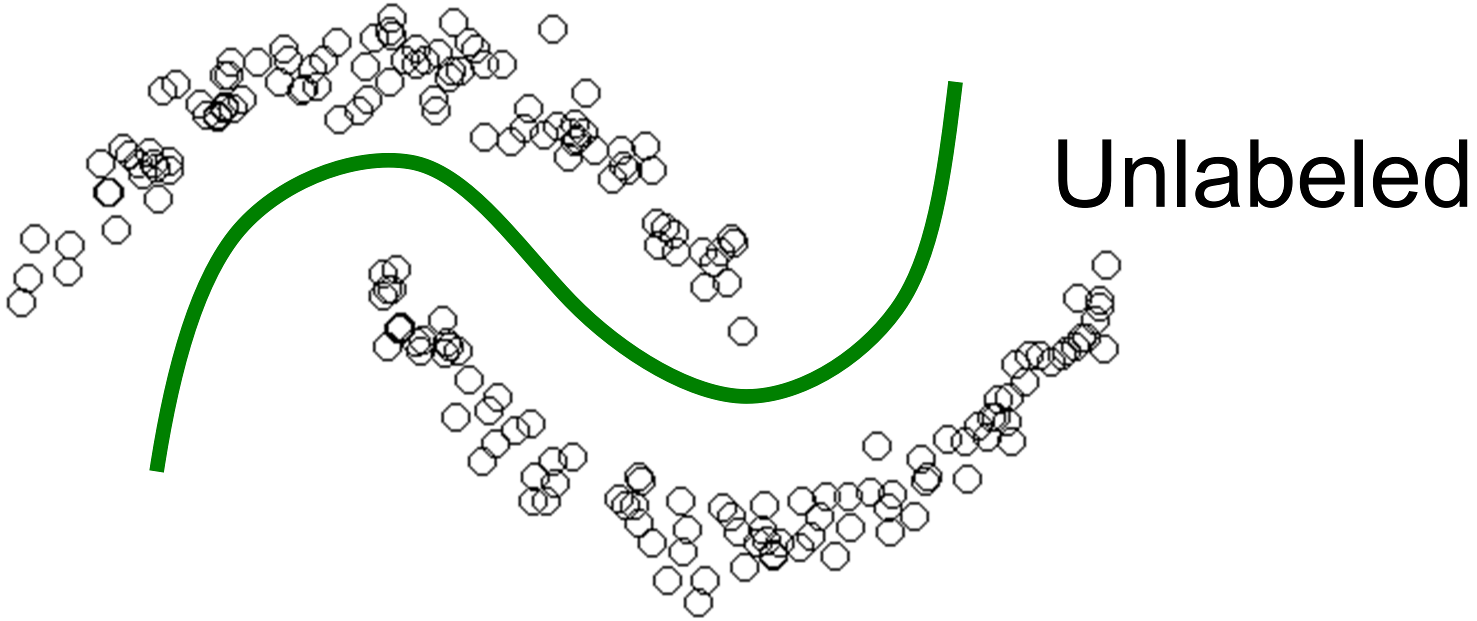
- Larger amount of labeled data yields better classification accuracy.
- Estimation error of the boundary decreases in order $1/\sqrt{n}$.



n : Number of labeled samples

Unsupervised Classification

- Gathering labeled data is costly. Let's use unlabeled data that are often cheap to collect:

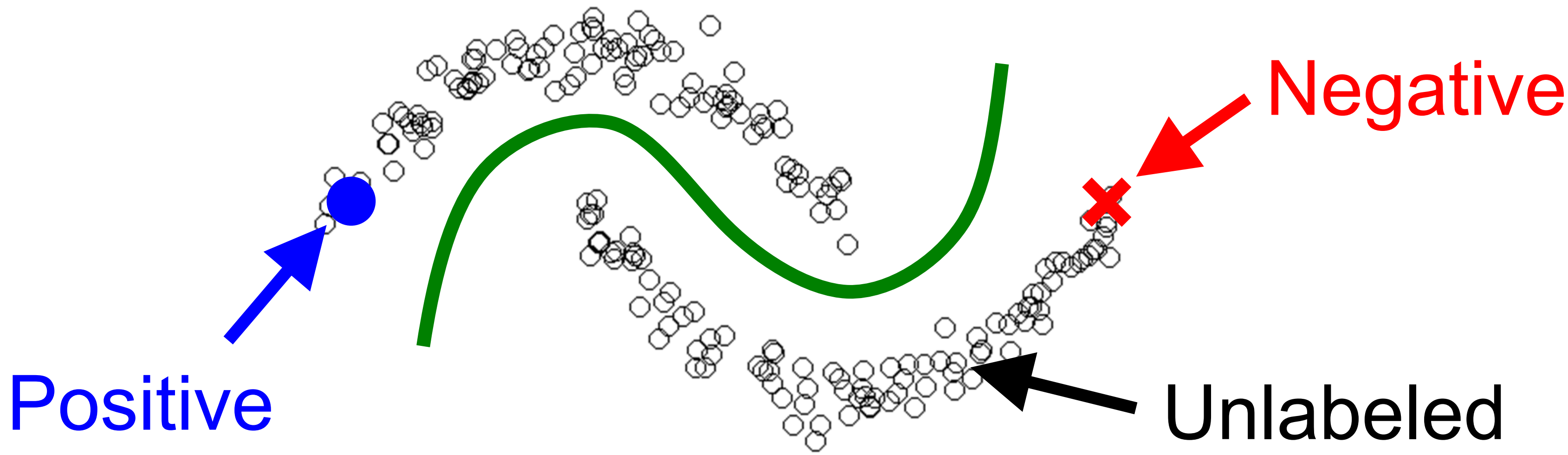


  - Unsupervised classification is typically clustering.
  - This works well only when each cluster corresponds to a class.

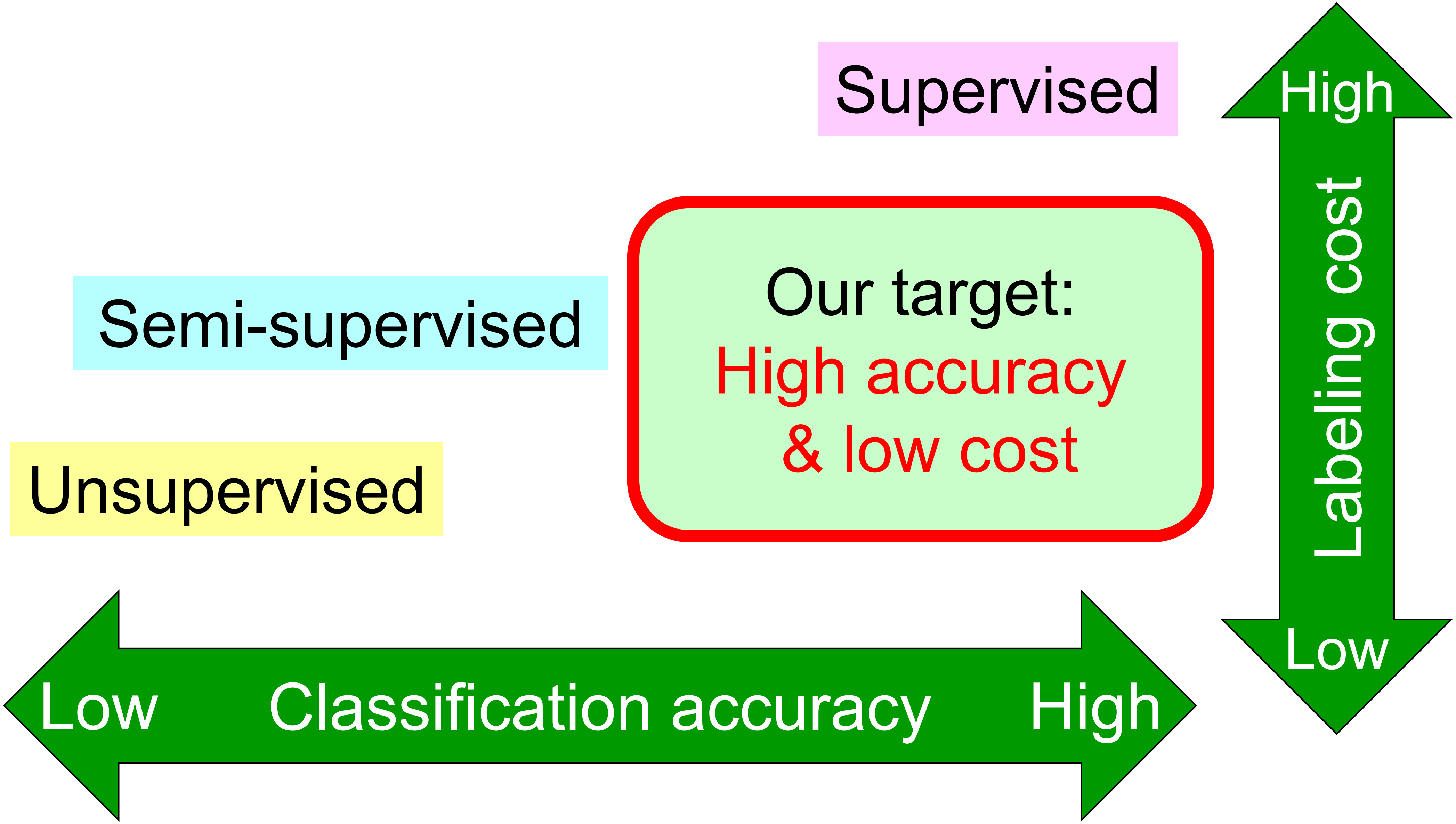
Semi-Supervised Classification

Chapelle, Schölkopf & Zien (MIT Press 2006) and many

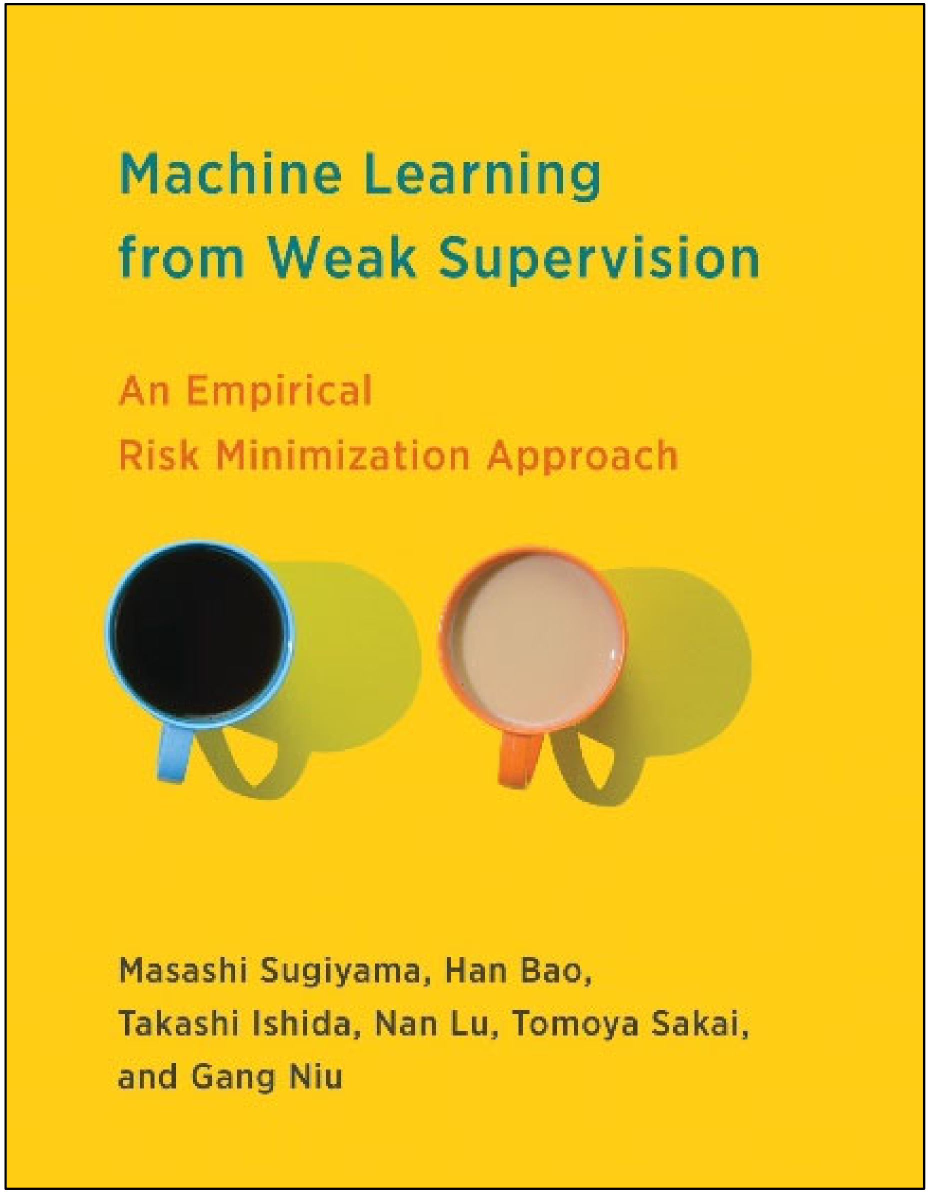
- Use a large number of unlabeled samples and a small number of labeled samples.
- Find a boundary along the cluster structure induced by unlabeled samples:
  - Sometimes very useful.
  - But not that different from unsupervised classification.



Classification of Classification



Textbook

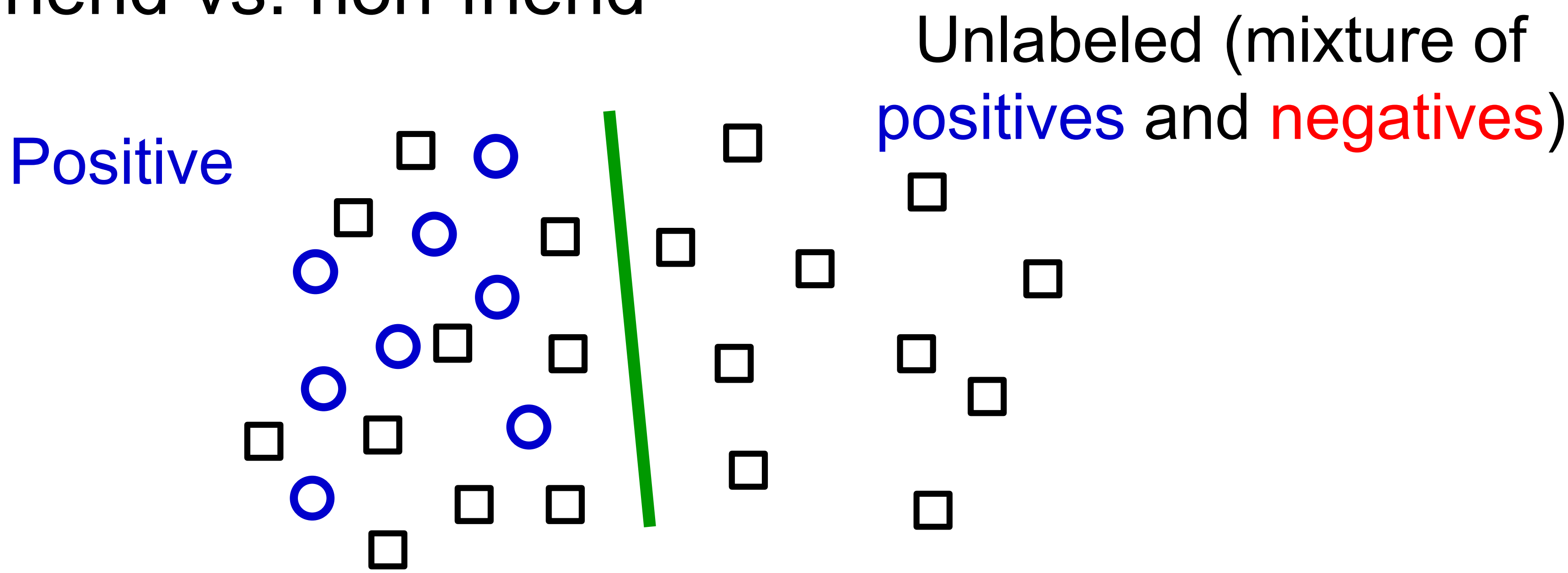


- Masashi Sugiyama, Han Bao, Takashi Ishida, Nan Lu, Tomoya Sakai, Gang Niu. Machine Learning from Weak Supervision: An Empirical Risk Minimization Approach, 320 pages, MIT Press, 2022.

PU Classification

du Plessis, Niu & Sugiyama (NIPS2014, ICML2015)
Niu, du Plessis, Sakai, Ma & Sugiyama (NIPS2016)
Kiryo, Niu, du Plessis & Sugiyama (NIPS2017)

- Only PU data is available; N data is missing:
  - Click vs. non-click
  - Friend vs. non-friend

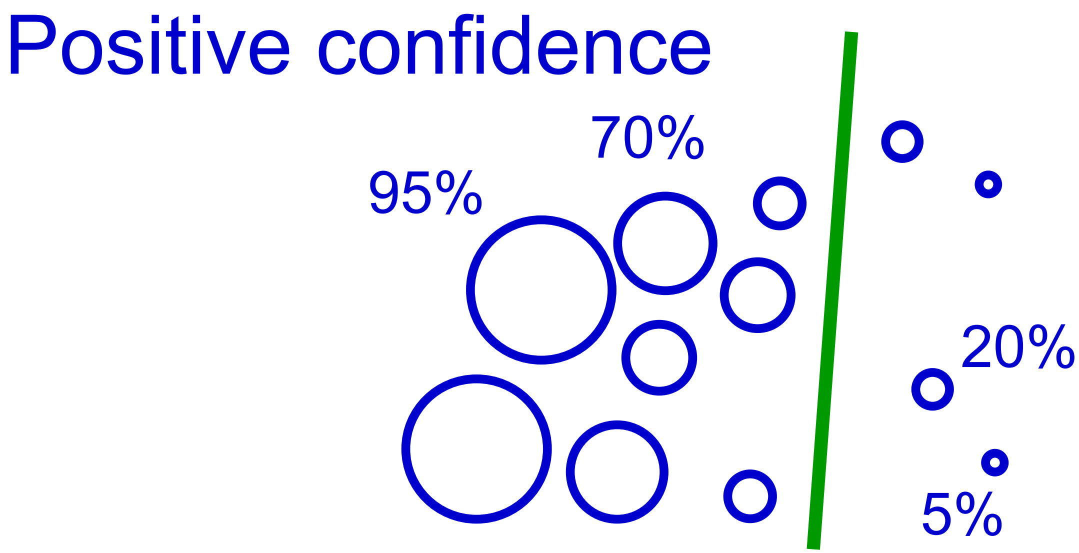


- From PU data, PN classifiers are trainable!

Pconf Classification

Ishida, Niu & Sugiyama (NeurIPS2018)

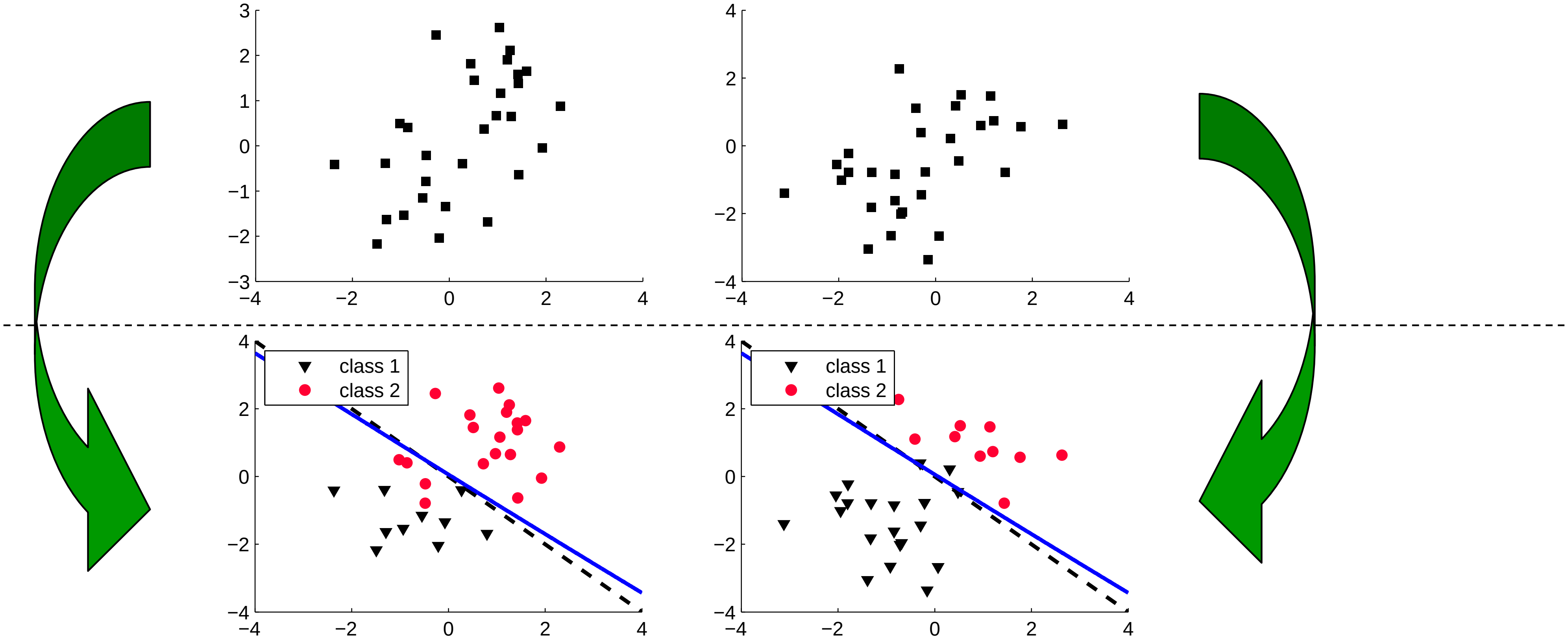
- Only P data is available, not U data:
  - Data from rival companies cannot be obtained.
  - Only positive results are reported (publication bias).
- "Only-P learning" is unsupervised.
- From Pconf data, PN classifiers are trainable!



UU Classification

du Plessis, Niu & Sugiyama (TAAI2013)
Nan, Niu, Menon & Sugiyama (ICLR2019)

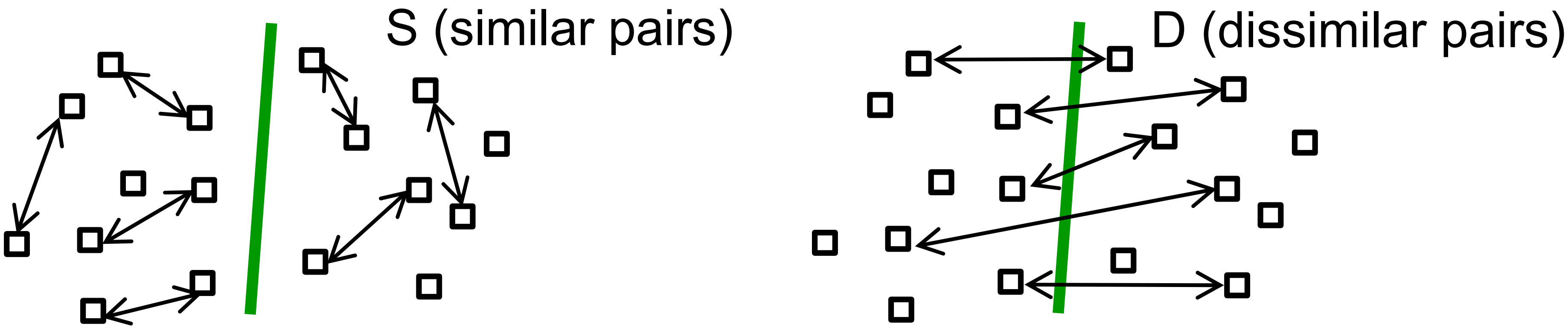
- From two sets of unlabeled data with different class priors, PN classifiers are trainable!



SDU Classification

Bao, Niu & Sugiyama (ICML2018)

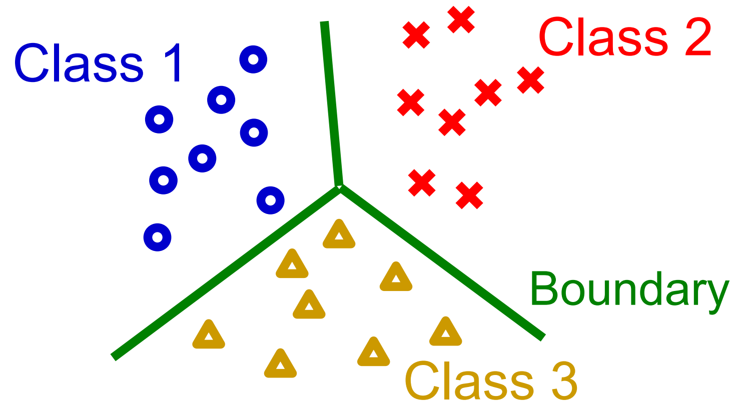
- Delicate classification (money, religion…):
  - Highly hesitant to directly answer questions.
  - Less reluctant to just say "same as him/her".
- From SU data, PN classifiers are trainable!



  - Learning from DU data is also possible.
  - Learning from SDU data is also possible.

Shimada, Bao, Sato & Sugiyama (NeCo2021)

Multiclass Methods



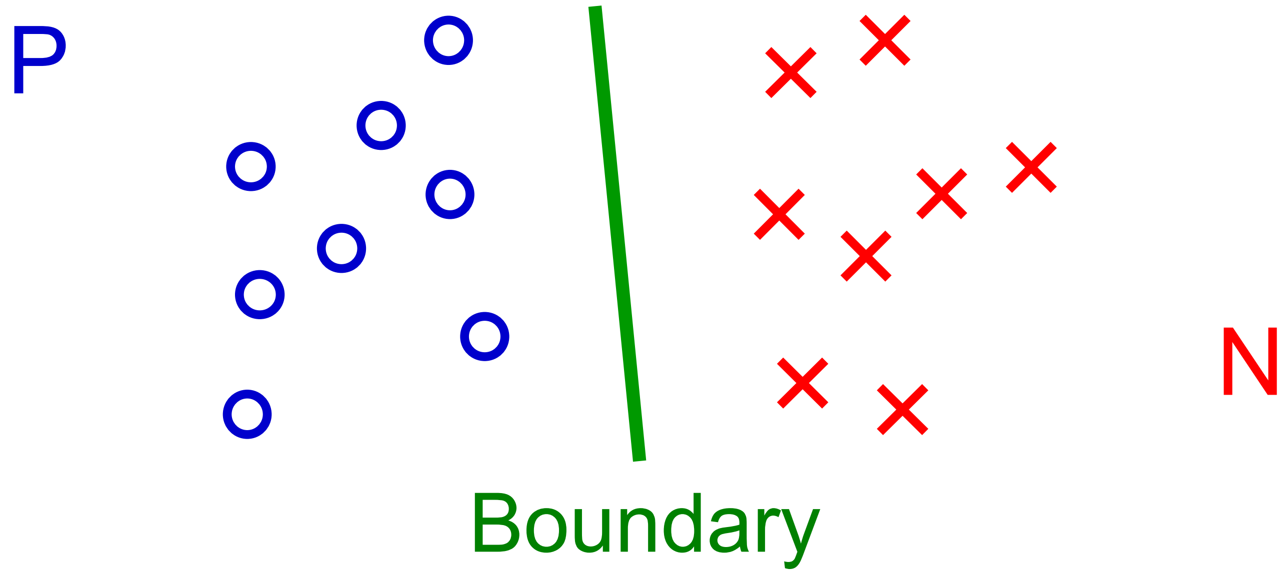
- Labeling patterns in multi-class problems is extremely painful.
  - Complementary labels: Specify a class that a pattern does not belong to ("not 1").
    Ishida, Niu, Hu & Sugiyama (NIPS2017)
    Ishida, Niu, Menon & Sugiyama (ICML2019)
  - Partial labels: Specify a subset of classes that contains the correct one ("1 or 2").
    Feng, Kaneko, Han, Niu, An & Sugiyama (ICML2020)
    Feng, Lv, Han, Xu, Niu, Geng, An & Sugiyama (NeurIPS2020)
  - Single-class confidence: One-class data with full confidence ("1 with 60%, 2 with 30%, and 3 with 10%")
    Cao, Feng, Shu, Xu, An, Niu & Sugiyama (arXiv2021)

Contents

- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- D: Dissimilar
- Comp: Complementary

PN Classification (Ordinary Supervised Classification)

- Labeled data: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y)$
  - Input $\boldsymbol{x} \in \mathbb{R}^d$: d-dimensional real vector
  - Output $y \in \{+1, -1\}$: Binary class label



Some Definitions

- **Classifier**: $f : \mathbb{R}^d \to \mathbb{R}$
  - Label prediction by $\widehat{y} = \mathrm{sign}(f(\boldsymbol{x}))$ (e.g., linear, additive, kernel, deep models).
- **Margin**: $m = yf(\boldsymbol{x})$ $\quad y \in \{+1, -1\}$
  - $m > 0 \implies \mathrm{sign}(f(\boldsymbol{x})) = y$
    → Classification is correct.
  - $m < 0 \implies \mathrm{sign}(f(\boldsymbol{x})) \neq y$
    → Classification is wrong.
- **Zero-one loss**: $\ell_{0/1}(m) = \frac{1}{2}(1 - \mathrm{sign}(m))$
  - 0 for correct prediction.
  - 1 for wrong prediction.

Classification Error and Empirical Approximation

- Classification error (expected zero-one loss over all test data):

$\mathbb{E}$: Expectation

$$R_{0/1}(f) = \mathbb{E}_{p(\boldsymbol{x},y)}\left[\ell_{0/1}\Big(yf(\boldsymbol{x})\Big)\right]$$

$$\ell_{0/1}(m) = \frac{1}{2}\left(1 - \mathrm{sign}\left(m\right)\right)$$

  - Our goal: Find a minimizer of $R_{0/1}(f)$.

- But this is impossible since $p(\boldsymbol{x}, y)$ is unknown:
  - Let's use samples: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y)$

i.i.d.: Independent and identically distributed

  - Empirical approximation:

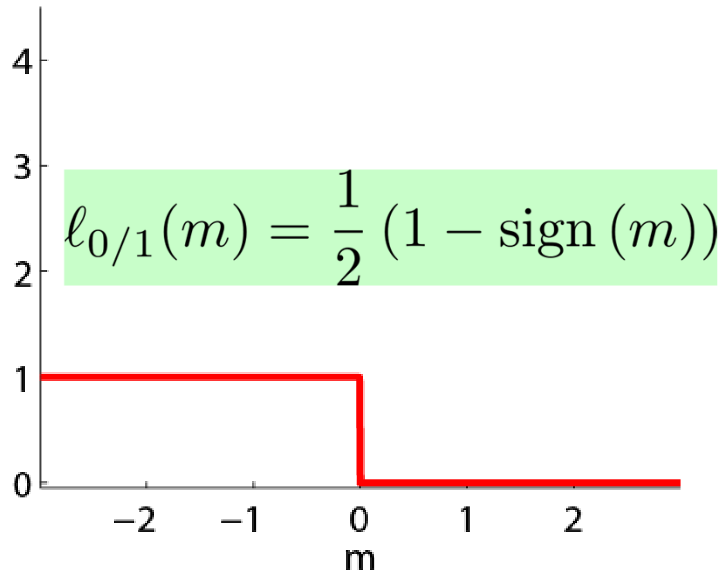
$$\widehat{R}_{0/1}(f) = \frac{1}{n}\sum_{i=1}^{n}\ell_{0/1}\Big(y_i f(\boldsymbol{x}_i)\Big) = R_{0/1}(f) + O_p\left(\frac{1}{\sqrt{n}}\right)$$

Minimization of Empirical Classification Error

$$\widehat{R}_{0/1}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell_{0/1}\Big(y_i f(\boldsymbol{x}_i)\Big)$$

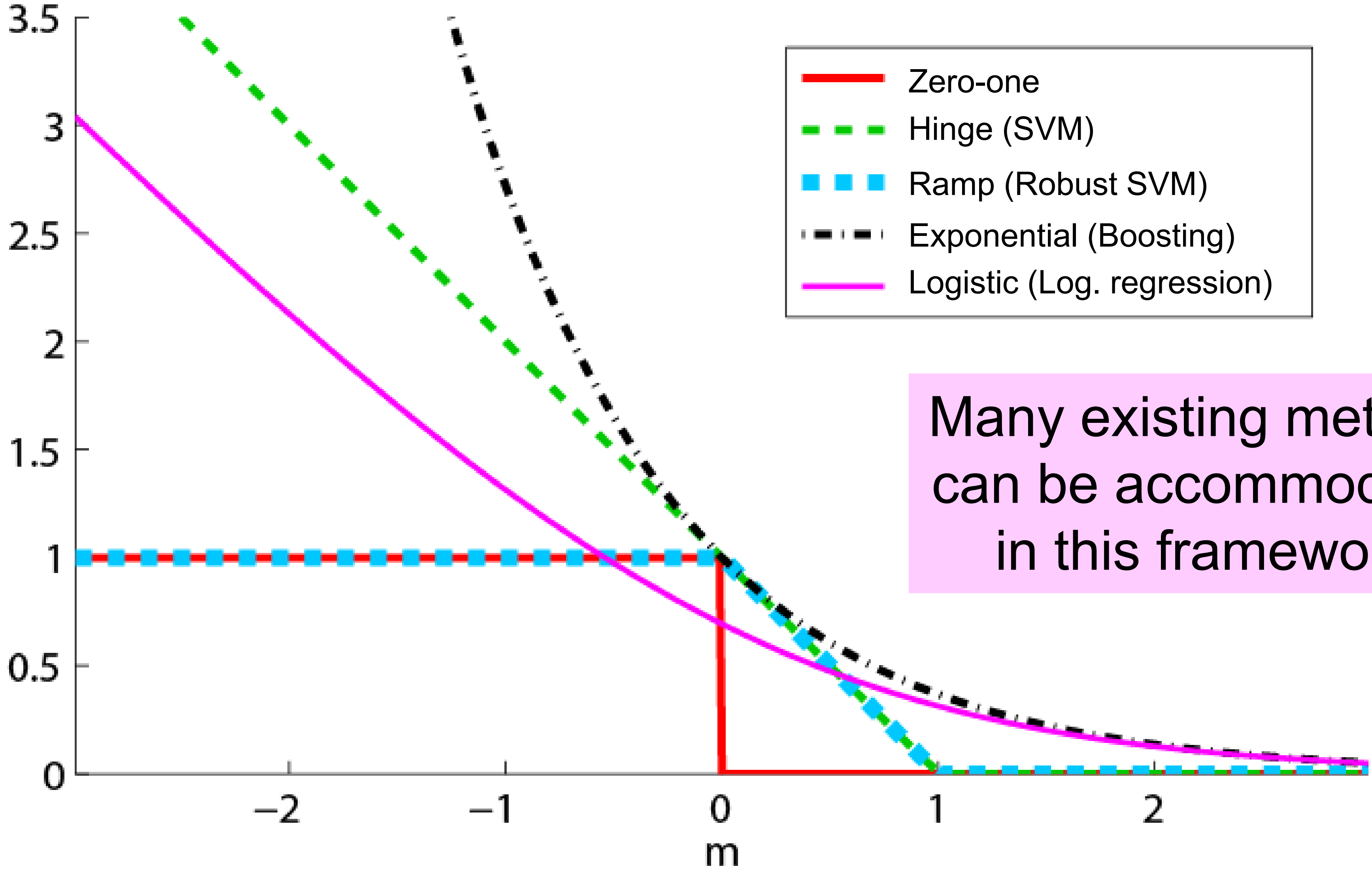
- However, minimization of $\widehat{R}_{0/1}(f)$ is NP-hard, due to discrete nature of $\ell_{0/1}$:
  - We may not be able to obtain a global minimizer in practice.

- Let's use a smoother loss!



Surrogate Loss

- Let's use a smoother loss as a surrogate:



Many existing methods can be accommodated in this framework!

PN Empirical Risk Minimization

- Classification risk for loss ℓ :

$$R(f) = \mathbb{E}_{p(\boldsymbol{x},y)}\left[\ell\Big(yf(\boldsymbol{x})\Big)\right]$$

- Empirical risk:
  - Expectation is approximated by sample average:

$$\widehat{R}_{\mathrm{PN}}(f) = \frac{1}{n}\sum_{i=1}^{n}\ell\Big(y_i f(\boldsymbol{x}_i)\Big) \quad = R(f) + O_p\left(\frac{1}{\sqrt{n}}\right)$$

$$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, y)$$

  - Minimize it within a certain model class (e.g., linear, additive, kernel, deep,…):

$$\widehat{f}_{\mathrm{PN}} = \underset{f}{\mathrm{argmin}}\, \widehat{R}_{\mathrm{PN}}(f)$$

Contents

- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- D: Dissimilar
- Comp: Complementary

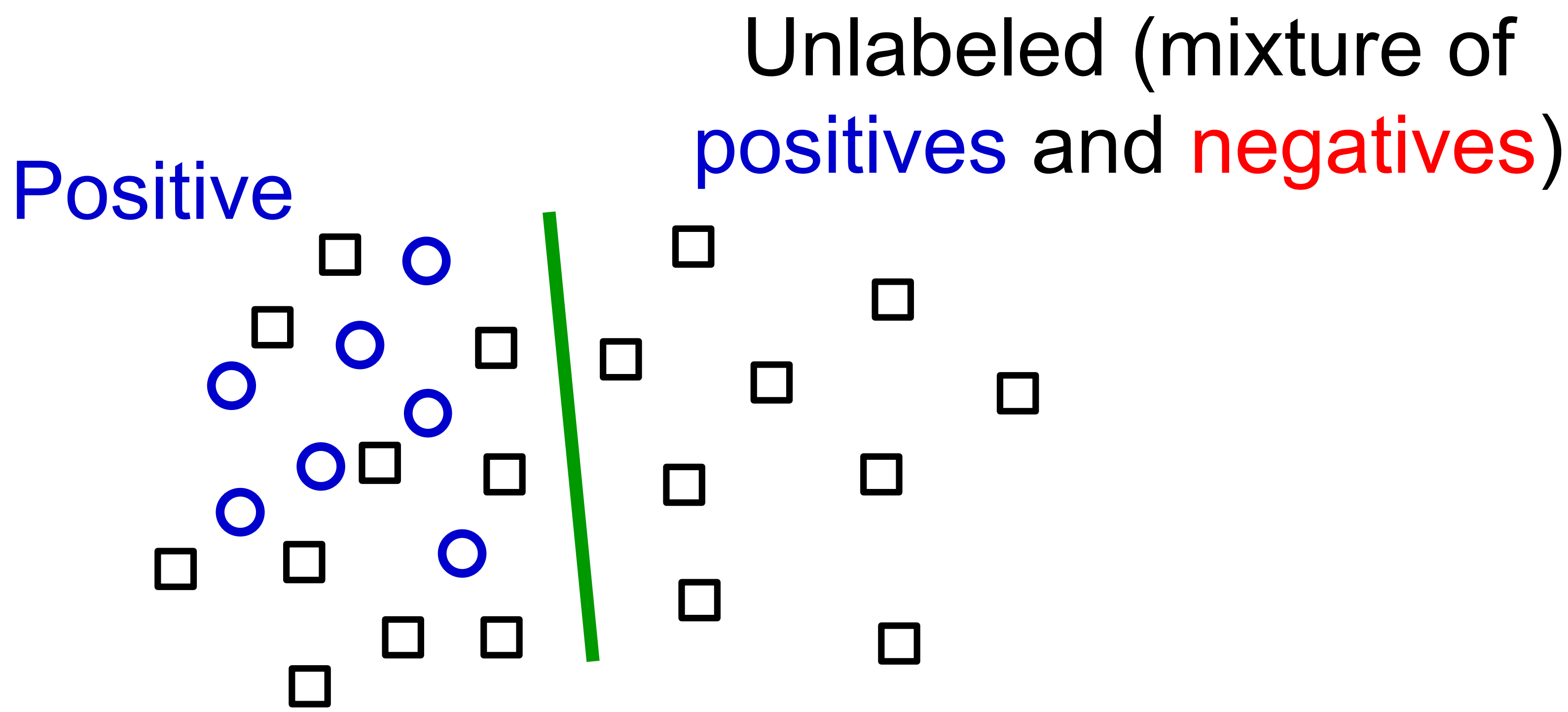
PU Classification: Setup

- Given: Positive and unlabeled samples

$$\{\boldsymbol{x}_i^{\mathrm{P}}\}_{i=1}^{n_{\mathrm{P}}} \overset{\mathrm{i.i.d.}}{\sim} p(\boldsymbol{x}|y=+1)$$

$$\{\boldsymbol{x}_i^{\mathrm{U}}\}_{i=1}^{n_{\mathrm{U}}} \overset{\mathrm{i.i.d.}}{\sim} p(\boldsymbol{x})$$

- Goal: Obtain a PN classifier



PN Risk Decomposition

- Risk of classifier f :

$$R(f) = \mathbb{E}_{p(\boldsymbol{x},y)}\left[\ell\Big(yf(\boldsymbol{x})\Big)\right]$$

ℓ : loss

$$= \underbrace{\pi\mathbb{E}_{p(\boldsymbol{x}|y=+1)}\left[\ell\Big(f(\boldsymbol{x})\Big)\right]}_{\text{Risk for P data}} + \underbrace{(1-\pi)\mathbb{E}_{p(\boldsymbol{x}|y=-1)}\left[\ell\Big(-f(\boldsymbol{x})\Big)\right]}_{\text{Risk for N data}}$$

$\pi = p(y=+1)$: Class-prior probability
(assumed known; can be estimated)

Scott & Blanchard (AISTATS2009)
Blanchard, Lee & Scott (JMLR2010)
du Plessis, Niu & Sugiyama (IEICE2014, MLJ2017)
Ramaswamy, Scott & Tewari (ICML2016)
Yao, Liu, Han, Gong, Niu, Sugiyama & Tao (ICLR2022)
https://www.ms.k.u-tokyo.ac.jp/sugi/slide/20211101_CIKM-LQ.pdf

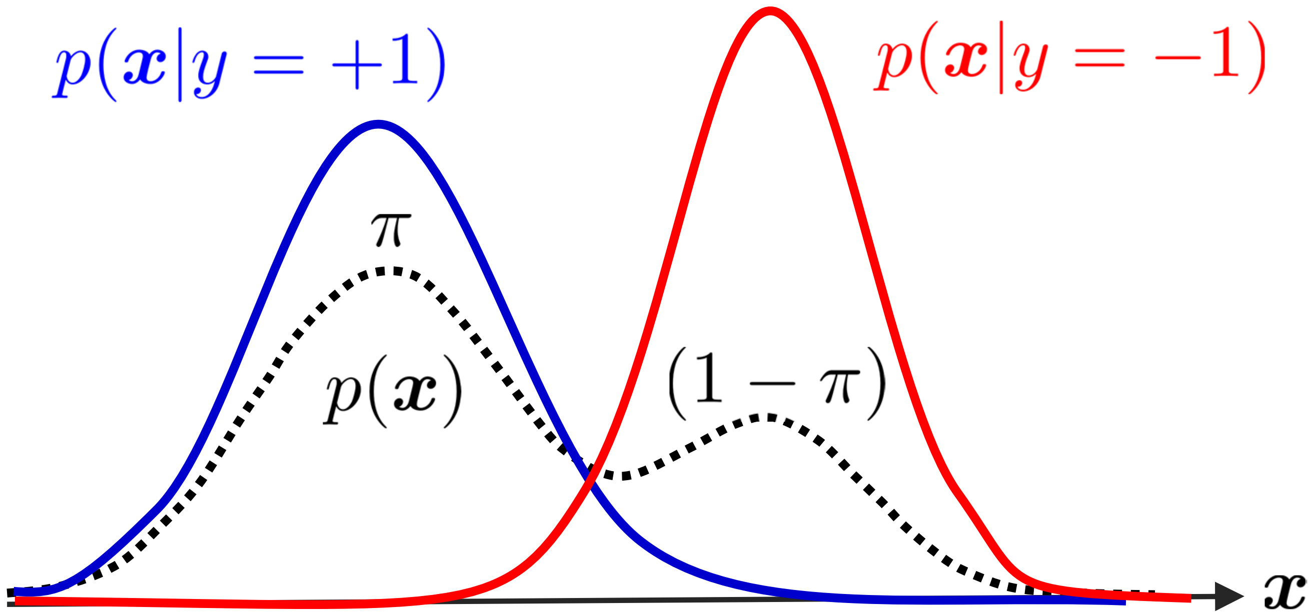
- Since we do not have N data in the PU setting, the risk cannot be directly estimated.

PU Risk Estimation

$$R(f) = \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\Big[\ell\Big(f(\boldsymbol{x})\Big)\Big] + (1-\pi)\mathbb{E}_{p(\boldsymbol{x}|y=-1)}\Big[\ell\Big(-f(\boldsymbol{x})\Big)\Big]$$

- U-density is a mixture of P- and N-densities:

$$p(\boldsymbol{x}) = \pi p(\boldsymbol{x}|y=+1) + (1-\pi)p(\boldsymbol{x}|y=-1)$$



PU Risk Estimation (cont.)

du Plessis, Niu & Sugiyama (ICML2015)

$$R(f) = \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\Big[\ell\Big(f(\boldsymbol{x})\Big)\Big] + (1-\pi)\mathbb{E}_{p(\boldsymbol{x}|y=-1)}\Big[\ell\Big(-f(\boldsymbol{x})\Big)\Big]$$

$$p(\boldsymbol{x}) = \pi p(\boldsymbol{x}|y=+1) + (1-\pi)p(\boldsymbol{x}|y=-1)$$

- This allow us to eliminate the N-density:

$$(1-\pi)p(\boldsymbol{x}|y=-1) = p(\boldsymbol{x}) - \pi p(\boldsymbol{x}|y=+1)$$

$$R(f) = \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\Big[\ell\Big(f(\boldsymbol{x})\Big)\Big] + \mathbb{E}_{p(\boldsymbol{x})}\Big[\ell\Big(-f(\boldsymbol{x})\Big)\Big] - \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\Big[\ell\Big(-f(\boldsymbol{x})\Big)\Big]$$

- Unbiased risk estimation is possible from PU data, just by replacing expectations by sample averages!

PU Empirical Risk Minimization

$$R(f) = \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\Big[\ell\Big(f(\boldsymbol{x})\Big)\Big] + \mathbb{E}_{p(\boldsymbol{x})}\Big[\ell\Big(-f(\boldsymbol{x})\Big)\Big] - \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\Big[\ell\Big(-f(\boldsymbol{x})\Big)\Big]$$

- Replacing expectations by sample averages gives an empirical risk:

$$\widehat{R}_{\mathrm{PU}}(f) = \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(f(\boldsymbol{x}_i^{\mathrm{P}})\Big) + \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\ell\Big(-f(\boldsymbol{x}_i^{\mathrm{U}})\Big) - \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(-f(\boldsymbol{x}_i^{\mathrm{P}})\Big)$$

$$\{\boldsymbol{x}_i^{\mathrm{P}}\}_{i=1}^{n_{\mathrm{P}}} \stackrel{\text{i.i.d.}}{\sim} p(\boldsymbol{x}|y=+1) \qquad \{\boldsymbol{x}_i^{\mathrm{U}}\}_{i=1}^{n_{\mathrm{U}}} \stackrel{\text{i.i.d.}}{\sim} p(\boldsymbol{x})$$

- Optimal convergence rate is attained:

Niu, du Plessis, Sakai, Ma & Sugiyama (NIPS2016)

$$R(\widehat{f}_{\mathrm{PU}}) - R(f^*) \leq C(\delta)\left(\frac{2\pi}{\sqrt{n_{\mathrm{P}}}} + \frac{1}{\sqrt{n_{\mathrm{U}}}}\right)$$

with probability $1-\delta$

$$\widehat{f}_{\mathrm{PU}} = \mathrm{argmin}_f\, \widehat{R}_{\mathrm{PU}}(f)$$

$$f^* = \mathrm{argmin}_f\, R(f)$$

$n_{\mathrm{P}}, n_{\mathrm{U}}$: # of P, U samples

Theoretical Comparison with PN

Niu, du Plessis, Sakai, Ma & Sugiyama (NIPS2016)

- Estimation error bounds for PU and PN:

$$R(\widehat{f}_{\mathrm{PU}}) - R(f^*) \leq C(\delta)\left(\frac{2\pi}{\sqrt{n_{\mathrm{P}}}} + \frac{1}{\sqrt{n_{\mathrm{U}}}}\right)$$

$$R(\widehat{f}_{\mathrm{PN}}) - R(f^*) \leq C(\delta)\left(\frac{\pi}{\sqrt{n_{\mathrm{P}}}} + \frac{1-\pi}{\sqrt{n_{\mathrm{N}}}}\right)$$

$$\widehat{f}_{\mathrm{PN}} = \operatorname*{argmin}_{f} \widehat{R}_{\mathrm{PN}}(f)$$

with probability $1-\delta$

$$\widehat{R}_{\mathrm{PN}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell\Big(y_i f(\boldsymbol{x}_i)\Big)$$

$n_{\mathrm{P}}, n_{\mathrm{N}}, n_{\mathrm{U}}$: # of P, N, U samples

- Comparison: PU bound is smaller than PN if

$$\frac{\pi}{\sqrt{n_{\mathrm{P}}}} + \frac{1}{\sqrt{n_{\mathrm{U}}}} < \frac{1-\pi}{\sqrt{n_{\mathrm{N}}}}$$

  - PU can be better than PN, provided many PU data!

Further Correction

$$R(f) = \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\left[\ell\Big(f(\boldsymbol{x})\Big)\right] + (1-\pi)\mathbb{E}_{p(\boldsymbol{x}|y=-1)}\left[\ell\Big(-f(\boldsymbol{x})\Big)\right]$$

Risk for P data

Risk for N data $R^{-}(f)$

- PU formulation: $p(\boldsymbol{x}) = \pi p(\boldsymbol{x}|y=+1) + (1-\pi)p(\boldsymbol{x}|y=-1)$

$$R^{-}(f) = \mathbb{E}_{p(\boldsymbol{x})}\left[\ell\Big(-f(\boldsymbol{x})\Big)\right] - \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\left[\ell\Big(-f(\boldsymbol{x})\Big)\right]$$

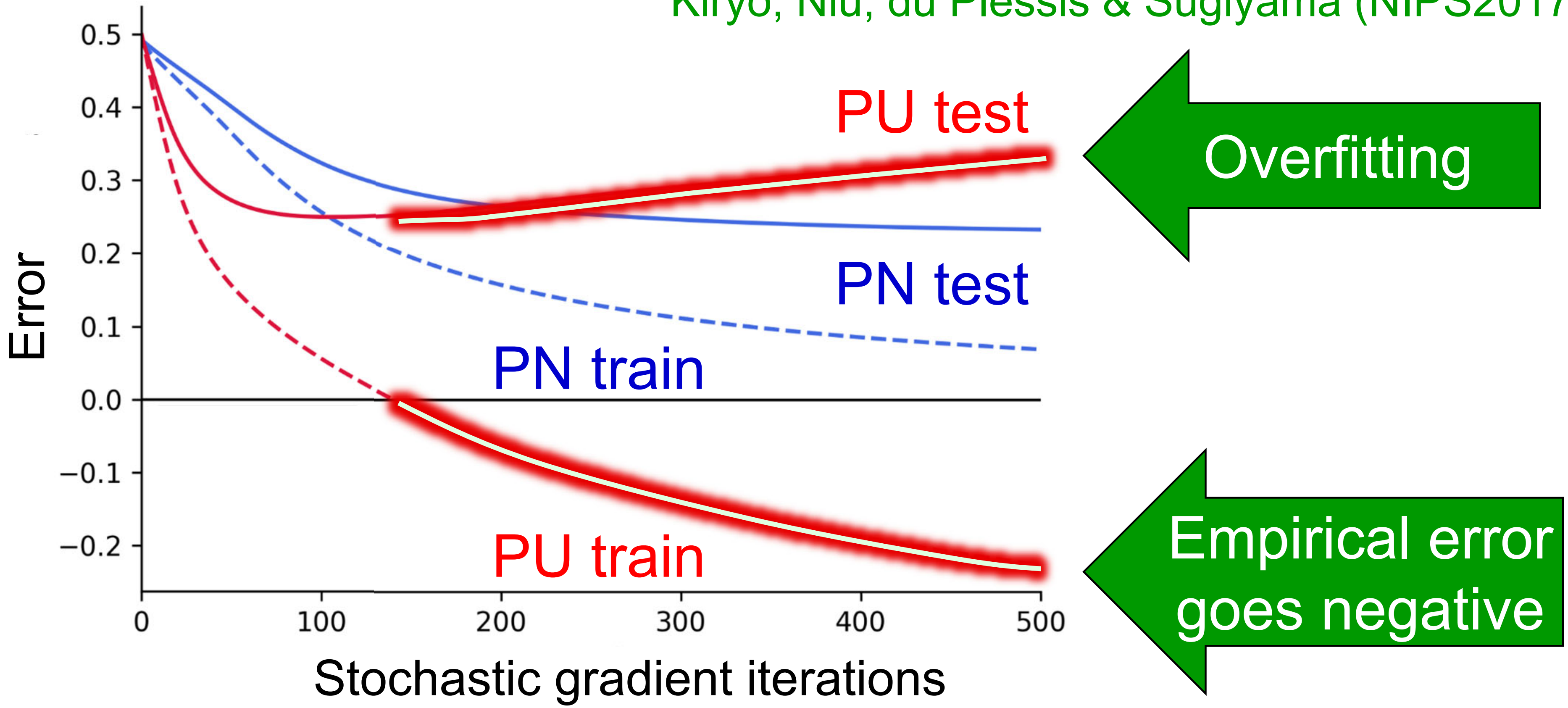
  - If $\ell(m) \geq 0, \ \forall m$, $R^{-}(f) \geq 0$.
  - However, its PU empirical approximation can be negative due to "difference of approximations".

$$\widehat{R}^{-}_{\mathrm{PU}}(f) = \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\ell\Big(-f(\boldsymbol{x}_i^{\mathrm{U}})\Big) - \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(-f(\boldsymbol{x}_i^{\mathrm{P}})\Big) \not\geq 0$$

  - This problem is more critical for flexible models such as deep nets.

Non-Negative PU Classification

Kiryo, Niu, du Plessis & Sugiyama (NIPS2017)



- We constrain the sample approximation term to be non-negative through back-prop training:

$$\widetilde{R}_{\mathrm{PU}}(f) = \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(f(\boldsymbol{x}_i^{\mathrm{P}})\Big) + \max\left\{0,\ \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\ell\Big(-f(\boldsymbol{x}_i^{\mathrm{U}})\Big) - \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(-f(\boldsymbol{x}_i^{\mathrm{P}})\Big)\right\}$$

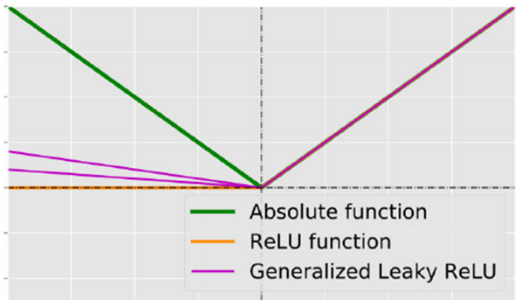
  - Now the risk estimator is biased. Is it really good?

Theoretical Analysis

Kiryo, Niu, du Plessis & Sugiyama (NIPS2017)

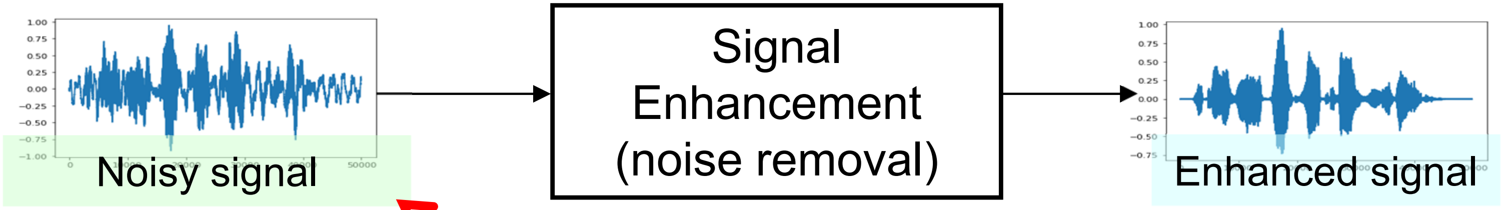
$$\widetilde{R}_{\mathrm{PU}}(f) = \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(f(\boldsymbol{x}_i^{\mathrm{P}})\Big)+\max\left\{0,\ \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}}\ell\Big(-f(\boldsymbol{x}_i^{\mathrm{U}})\Big)-\frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(-f(\boldsymbol{x}_i^{\mathrm{P}})\Big)\right\}$$

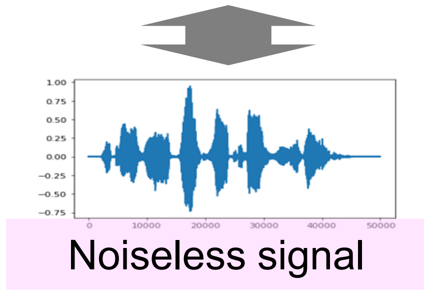
- $\widetilde{R}_{\mathrm{PU}}(f)$ is still consistent and its bias decreases exponentially: $\mathcal{O}(e^{-n_{\mathrm{P}}-n_{\mathrm{U}}})$ $n_{\mathrm{P}}, n_{\mathrm{U}}$: # of P, U samples
  - In practice, we can ignore the bias of $\widetilde{R}_{\mathrm{PU}}(f)$!
- Mean-squared error of $\widetilde{R}_{\mathrm{PU}}(f)$ is not more than the original one.
  - In practice, $\widetilde{R}_{\mathrm{PU}}(f)$ is more reliable!
- Risk of $\mathrm{argmin}_f\,\widetilde{R}_{\mathrm{PU}}(f)$ for linear models attains optimal convergence rate: $\mathcal{O}_p\left(\frac{1}{\sqrt{n_{\mathrm{P}}}}+\frac{1}{\sqrt{n_{\mathrm{U}}}}\right)$
  - Learned function is optimal.
- Extension to leaky-ReLU: Lu, Zhang, Niu & Sugiyama (AISTATS2020)
  - Corresponding to gradient ascent.



Signal Enhancement

Ito & Sugiyama (ICASSP2023)

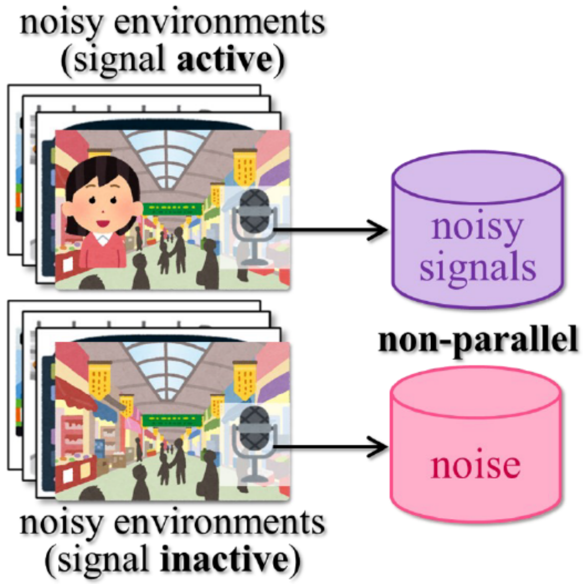




- **Existing method:** Use noisy/noiseless parallel training data
  - In practice, use synthetic data
    → Do not generalize well in reality.

- **Proposed method:** Use non-parallel noisy signal and noise.

	Methods	SI-SNRi [dB]
Non-parallel	Proposed	14.62 (0.20)
Non-parallel	MixIT Wisdom+ (NeurIPS2020)	12.19 (4.50)
Parallel	Supervised	15.86 (1.28)



Contents

1. Introduction
2. PN Classification
3. PU Classification
4. Pconf Classification
5. UU Classification
6. SDU Classification
7. Comp. Classification
8. Summary

- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- D: Dissimilar
- Comp: Complementary

Pconf Classification: Setup

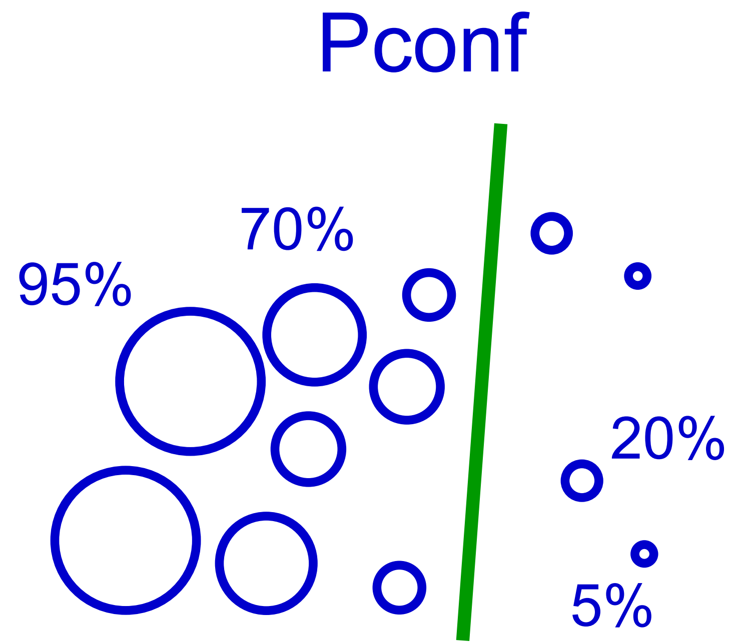
Ishida, Niu & Sugiyama (NeurIPS2018)

- Given: Positive-confidence samples

$$\{(\boldsymbol{x}_i, r_i)\}_{i=1}^{n}$$

  - Positive patterns: $\{\boldsymbol{x}_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}|y=+1)$
  - Their confidence: $r_i = P(y=+1|\boldsymbol{x}_i)$

- Goal: Obtain a PN classifier



Pconf Risk Estimation

- Classification risk: $R(f) = \mathbb{E}_{p(\boldsymbol{x},y)}\left[\ell\Big(yf(\boldsymbol{x})\Big)\right]$
- Naïve "confidence-weighting" is not correct.

$$R(f) \neq \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\left[r(\boldsymbol{x})\ell\Big(f(\boldsymbol{x})\Big) + (1-r(\boldsymbol{x}))\ell\Big(-f(\boldsymbol{x})\Big)\right]$$

$$r(\boldsymbol{x}) = P(y=+1|\boldsymbol{x})$$

- Correct form is given by importance sampling:

$$R(f) = \pi\mathbb{E}_{p(\boldsymbol{x}|y=+1)}\left[\ell\Big(f(\boldsymbol{x})\Big) + \frac{1-r(\boldsymbol{x})}{r(\boldsymbol{x})}\ell\Big(-f(\boldsymbol{x})\Big)\right]$$

resulting in an empirical risk:

$$\widehat{R}_{\mathrm{Pconf}}(f) \propto \sum_{i=1}^{n}\left[\ell\Big(f(\boldsymbol{x}_i)\Big) + \frac{1-r_i}{r_i}\ell\Big(-f(\boldsymbol{x}_i)\Big)\right]$$

$$\{\boldsymbol{x}_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}|y=+1) \quad r_i = P(y=+1|\boldsymbol{x}_i)$$

Contents

- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- D: Dissimilar
- Comp: Complementary

UU Classification: Setup

du Plessis, Niu & Sugiyama (TAAI2013)
Lu, Niu, Menon & Sugiyama (ICLR2019)
Lu, Zhang, Niu & Sugiyama (AISTATS2020)

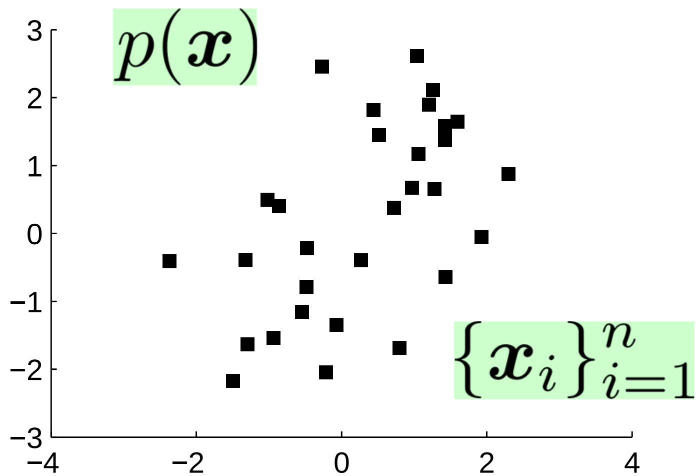
- **Given:** Two sets of unlabeled data

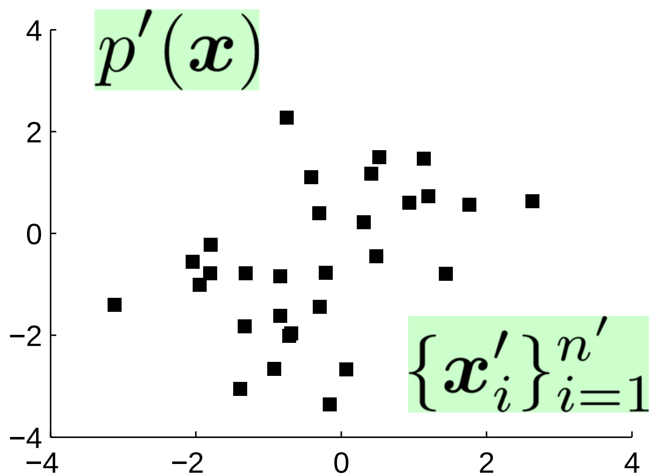
$$\{\boldsymbol{x}_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}) \qquad \{\boldsymbol{x}'_i\}_{i=1}^{n'} \overset{\text{i.i.d.}}{\sim} p'(\boldsymbol{x})$$

- **Assumption:** Only class-priors are different

$$p(y) \neq p'(y) \qquad p(\boldsymbol{x}|y) = p'(\boldsymbol{x}|y)$$

- **Goal:** Obtain a PN classifier





Optimal UU Classifier

du Plessis, Niu & Sugiyama (TAAI2013)

- Sign of the difference of class-posteriors:

$$g(\boldsymbol{x}) = \mathrm{sign}[p(y=+1|\boldsymbol{x}) - p(y=-1|\boldsymbol{x})]$$

- Under uniform test class-prior,

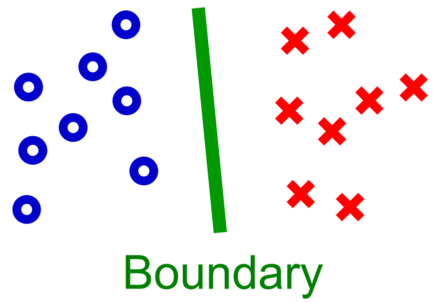
$$g(\boldsymbol{x}) = C\mathrm{sign}[p(\boldsymbol{x}) - p'(\boldsymbol{x})]$$

$$C = \mathrm{sign}[p(y=+1) - p'(y=+1)]$$

- Sign of C is unknown, but just knowing

$$\mathrm{sign}[p(\boldsymbol{x}) - p'(\boldsymbol{x})]$$

still allows optimal separation!



UU Risk Estimation

Lu, Niu, Menon & Sugiyama (ICLR2019)

- For
  - uniform test class-prior: $\pi = 1/2$
  - symmetric loss: $\ell(m) + \ell(-m) = \text{Const.}$

the classification risk can be expressed as

$$R(f) = \mathbb{E}_{p(\boldsymbol{x},y)}\left[\ell\Big(yf(\boldsymbol{x})\Big)\right]$$

$$\propto \mathbb{E}_{p(\boldsymbol{x})}\left[\ell\Big(f(\boldsymbol{x})\Big)\right] + \mathbb{E}_{p'(\boldsymbol{x}')}\left[\ell\Big(-f(\boldsymbol{x}')\Big)\right] + \text{Const.}$$

resulting an empirical risk (up to label flip):

$$\widehat{R}_{\mathrm{UU}}(f) \propto \frac{1}{n}\sum_{i=1}^{n}\ell\Big(f(\boldsymbol{x}_i)\Big) + \frac{1}{n'}\sum_{i=1}^{n'}\ell\Big(-f(\boldsymbol{x}'_i)\Big)$$

$$\{\boldsymbol{x}_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}) \qquad \{\boldsymbol{x}'_i\}_{i=1}^{n'} \overset{\text{i.i.d.}}{\sim} p'(\boldsymbol{x})$$

Extension to UUU…

$m\ (\geq 2)$

- $\mathrm{U^m}$ classification: m U sets $\{\boldsymbol{x}_i^{(j)}\}_{i=1,j=1}^{n_j,m}$ are given.
- Apply UU for pairs of U sets: Scott & Zhang (NeurIPS2020)
  - However, it is computationally expensive.
- Surrogate set classification:

  Lu, Lei, Niu, Sato & Sugiyama (ICML2021)

  - Learn an m-class classifier $\boldsymbol{f}(\boldsymbol{x})$ that probabilistically assigns the dataset ID to each sample.

$$\{\boldsymbol{x}_i^{(j)}, \bar{y}_i^{(j)} = j\}_{i=1,j=1}^{n_j,m} \Longrightarrow \boldsymbol{p}(\bar{y}|\boldsymbol{x}) \approx \boldsymbol{f}(\boldsymbol{x})$$

  - It can be deterministically converted to the classifier that assigns PN labels to each sample.

$$\boldsymbol{p}(y|\boldsymbol{x}) \approx \boldsymbol{T}(\boldsymbol{f}(\boldsymbol{x}))$$

Contents

1. Introduction
2. PN Classification
3. PU Classification
4. Pconf Classification
5. UU Classification
6. SDU Classification
7. Comp. Classification
8. Summary

- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- D: Dissimilar
- Comp: Complementary

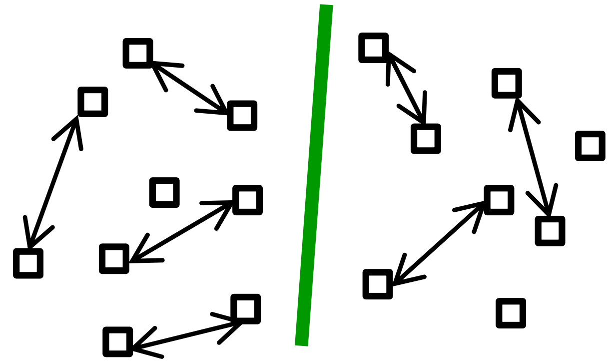
SU Classification

Bao, Niu & Sugiyama (ICML2018)

- Given: Similar and unlabeled samples

$$\{(\boldsymbol{x}_i, \boldsymbol{x}'_i)\}_{i=1}^{n_{\mathrm{S}}} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}, \boldsymbol{x}' | y = y')$$

$$\{\boldsymbol{x}_i^{\mathrm{U}}\}_{i=1}^{n_{\mathrm{U}}} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x})$$



- Goal: Obtain a PN classifier
- This is a special case of UU classification:

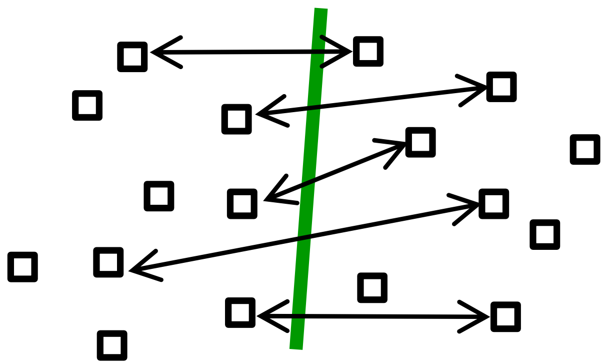
$$p(y = +1) = \pi^2/(2\pi^2 - 2\pi + 1)$$

$$p'(y = +1) = \pi$$

Extensions to DU/SD/SDU

Shimada, Bao, Sato & Sugiyama (NeCo2021)

- DU and SD classification are also special cases of UU classification:
  - DU: $p(y=+1)=1/2$
    $p'(y=+1)=\pi$
  - SD: $p(y=+1)=\pi^2/(2\pi^2-2\pi+1)$
    $p'(y=+1)=1/2$



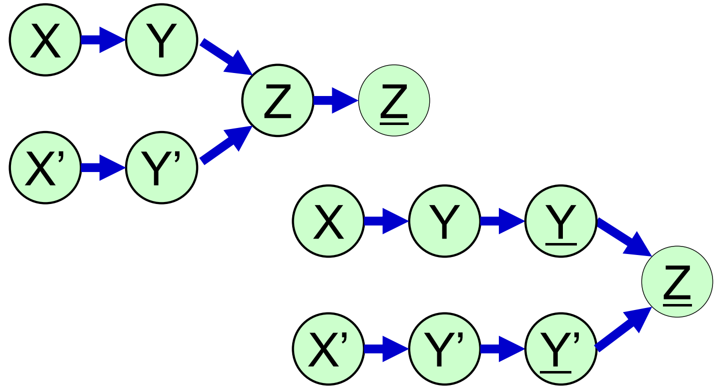
- SDU classification is also possible by combining DU/SU/SD classification (in the same way as PNU classification).

Further Extensions

- **Noisy SD**: Two types of noise:

  Dan, Bao & Sugiyama (ECMLPKDD2021)

  - Pairing corruption noise: Pairwise labels (S/D) are noisy.
  - Labeling corruption noise: Latent class labels (P/N) are noisy.



- **Similar-confidence** (Sconf):

  Cao, Feng, Xu, An, Niu & Sugiyama (ICML2021)

  - Similar pairs with confidence. $p(\boldsymbol{x}, \boldsymbol{x}' | y = y')$

- **Pairwise confidence comparison**:

  Feng, Shu, Lu, Han, Xu, Niu, An & Sugiyama (ICML2021)

  - Sample pairs with one having larger Pconf than the other. $p(y = +1 | \boldsymbol{x}) > p(y = +1 | \boldsymbol{x}')$

- **Confidence difference**:

  Wang, Feng, Jiang, Niu, Zhang & Sugiyama (NeurIPS2023)

$$c(\boldsymbol{x}, \boldsymbol{x}') = p(y = +1 | \boldsymbol{x}) - p(y = +1 | \boldsymbol{x}')$$

Contents

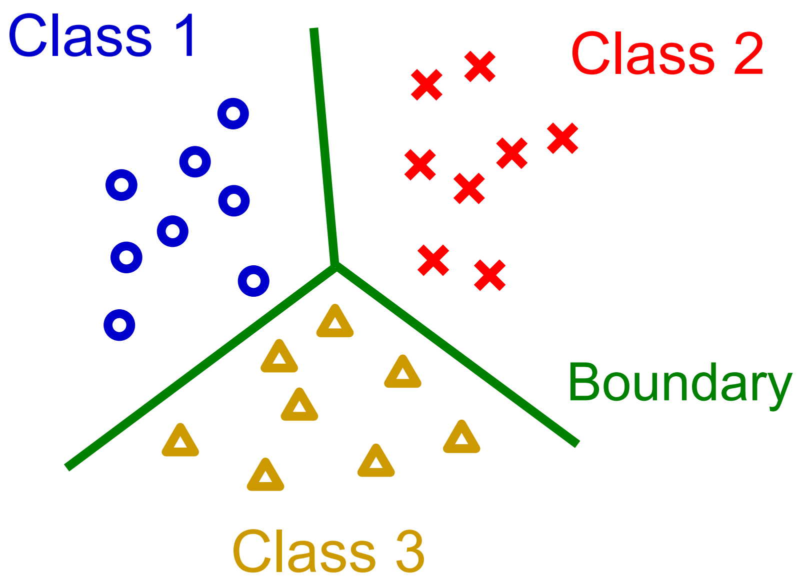
1. Introduction
2. PN Classification
3. PU Classification
4. Pconf Classification
5. UU Classification
6. SDU Classification
7. Comp. Classification
8. Summary

- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- D: Dissimilar
- Comp: Complementary

Complementary Labels

Ishida, Niu, Hu & Sugiyama (NIPS2017)
Ishida, Niu, Menon & Sugiyama (ICML2019)

- Labeling patterns in multi-class problems:
  - Selecting a correct class from a long list of candidate classes is extremely painful.
- Complementary labels:
  - Specify a class that a pattern does not belong to.
  - This is much easier and faster to perform!
- From complementary labels, classifiers are trainable!



$1/\sqrt{n}$

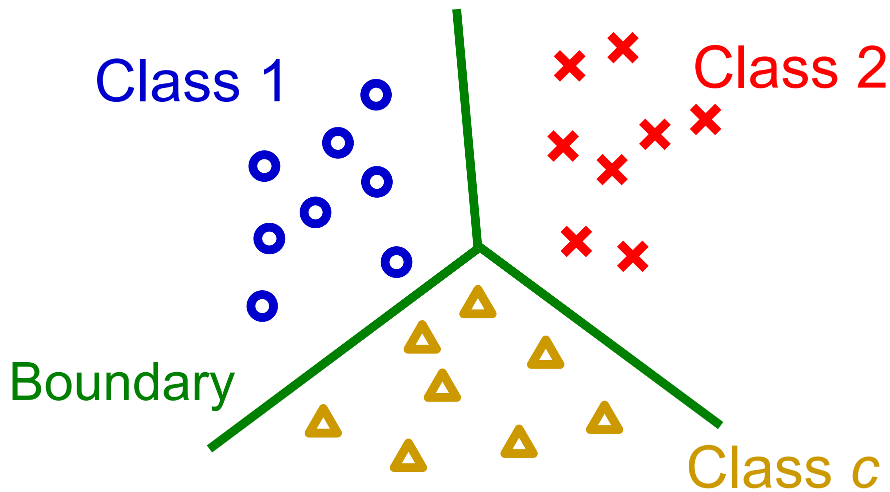
Complementary Classification

- **Given:** Complementarily labeled data

$$\{(\boldsymbol{x}_i, \bar{y}_i)\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} \bar{p}(\boldsymbol{x}, \bar{y}) \qquad \bar{p}(\boldsymbol{x}, \bar{y}) = \frac{1}{c-1} \sum_{y \neq \bar{y}} p(\boldsymbol{x}, y)$$

  - Pattern x does not belong to class $\bar{y} \in \{1, 2, \ldots, c\}$.

- **Goal**: Obtain a multiclass classifier



Multi-Class Classification

- c-class classifier: $f(\boldsymbol{x}) = \underset{y \in \{1,\ldots,c\}}{\operatorname{argmax}} \; g_y(\boldsymbol{x})$

  $g_y(\boldsymbol{x})$: one-vs-rest classifier for y

- c-class loss: $L\big(y, \boldsymbol{g}(\boldsymbol{x})\big)$ $\quad \boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), \ldots, g_c(\boldsymbol{x}))^\top$
  - One-versus-rest:

$$L_{\mathrm{OVR}}\big(y, \boldsymbol{g}(\boldsymbol{x})\big) = \ell\big(g_y(\boldsymbol{x})\big) + \frac{1}{c-1} \sum_{y' \neq y} \ell\big(-g_{y'}(\boldsymbol{x})\big)$$

  - Pairwise comparison:

$$L_{\mathrm{PC}}\big(y, \boldsymbol{g}(\boldsymbol{x})\big) = \sum_{y' \neq y} \ell\big(g_y(\boldsymbol{x}) - g_{y'}(\boldsymbol{x})\big)$$

- c-class classification risk:

$$R(\boldsymbol{g}) = \mathbb{E}_{p(\boldsymbol{x},y)}\Big[L\big(y, \boldsymbol{g}(\boldsymbol{x})\big)\Big]$$

Complementary Risk Estimation

Ishida, Niu, Menon & Sugiyama (ICML2019)

$$R(\boldsymbol{g}) = \mathbb{E}_{p(\boldsymbol{x},y)}\left[L\Big(y, \boldsymbol{g}(\boldsymbol{x})\Big)\right]$$

- Risk can be equivalently expressed as

$$R(\boldsymbol{g}) = \mathbb{E}_{\bar{p}(\boldsymbol{x},\bar{y})}\left[\bar{L}\Big(\bar{y}, \boldsymbol{g}(\boldsymbol{x})\Big)\right]$$

  - Complementary loss:

$$\bar{L}\Big(\bar{y}, \boldsymbol{g}(\boldsymbol{x})\Big) = -(c-1)L\Big(\bar{y}, \boldsymbol{g}(\boldsymbol{x})\Big) + \sum_{y=1}^{c} L\Big(y, \boldsymbol{g}(\boldsymbol{x})\Big)$$

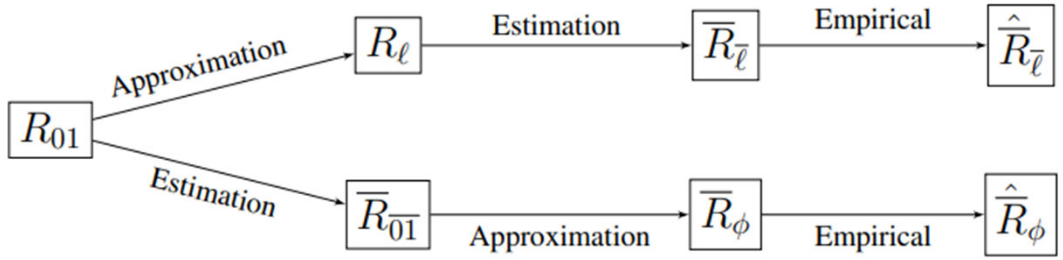
- Empirical risk estimation is possible from complementary data!

$$\widehat{R}_{\mathrm{Comp}}(\boldsymbol{g}) = \frac{1}{n}\sum_{i=1}^{n}\bar{L}\Big(\bar{y}_i, \boldsymbol{g}(\boldsymbol{x}_i)\Big) \qquad \{(\boldsymbol{x}_i, \bar{y}_i)\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} \bar{p}(\boldsymbol{x}, \bar{y})$$

Generalizations

- From unbiased risk estimation to surrogate complementary loss:
  - Surrogate approximation later.

Chou, Niu, G., Lin & Sugiyama (ICML2020)



- Multiple complementary labels (=partial labels):
  - Consider the size of complementary sets.

Feng, Kaneko, Han, Niu, An & Sugiyama (ICML2020)

$$\bar{p}(\boldsymbol{x}, \bar{Y}) = \sum_{j=1}^{k-1} p(s=j)\bar{p}(\boldsymbol{x}, \bar{Y} \mid s=j)$$

$$\bar{p}(\boldsymbol{x}, \bar{Y} \mid s=j) := \begin{cases} \frac{1}{\binom{k-1}{j}} \sum_{y \notin \bar{Y}} p(\boldsymbol{x}, y), & \text{if } |\bar{Y}| = j, \\ 0, & \text{otherwise.} \end{cases}$$

- Release from the uniform assumption:
  - Selected completely at random.

Wang, Ishida, Zhang, Niu & Sugiyama (arXiv2023)

$$p\left(k \in \bar{Y} \mid \boldsymbol{x}, k \in \mathcal{Y}\backslash\{y\}\right) = p\left(k \in \bar{Y} \mid k \in \mathcal{Y}\backslash\{y\}\right) = c_k$$

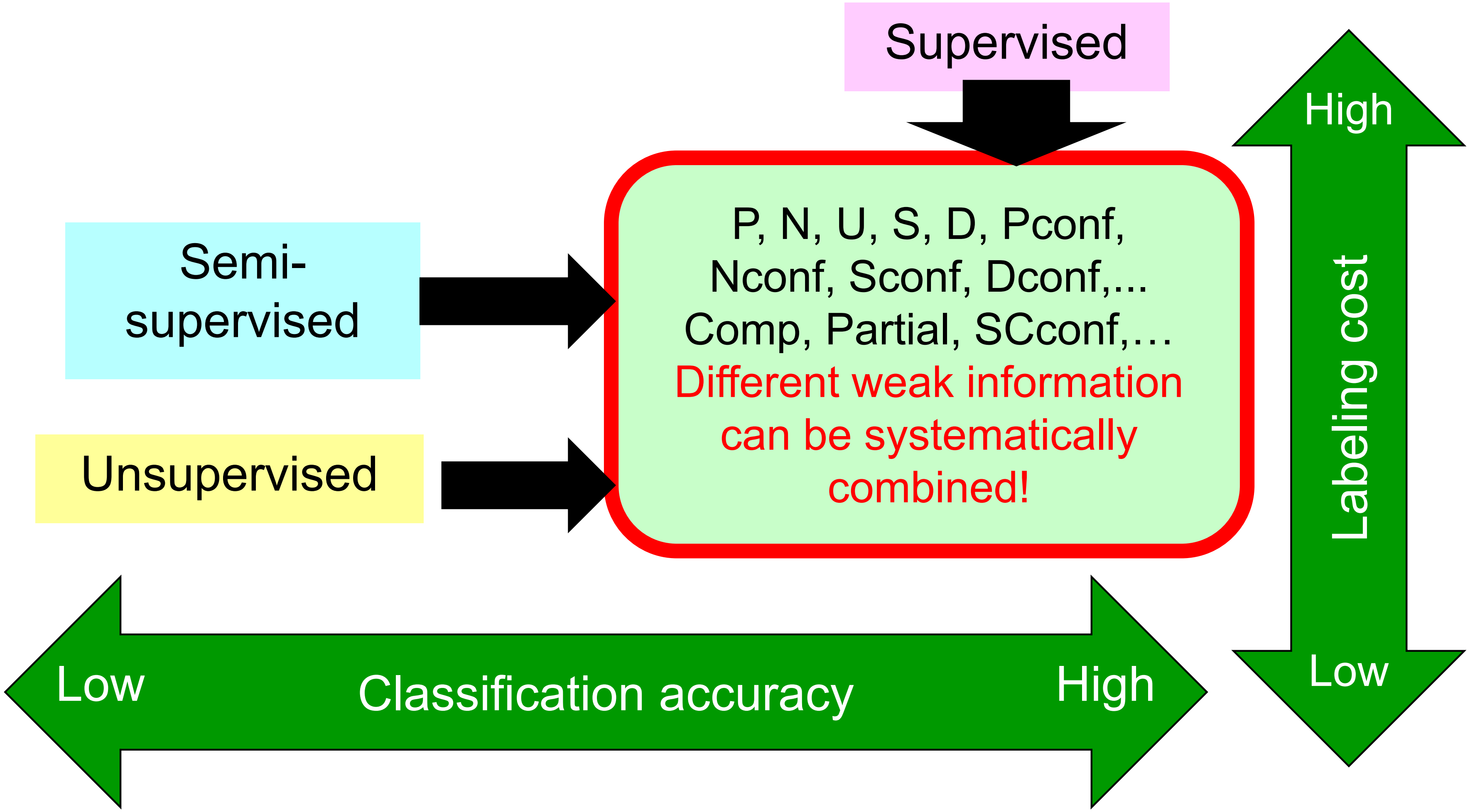
Contents

1. Introduction
2. PN Classification
3. PU Classification
4. Pconf Classification
5. UU Classification
6. SDU Classification
7. Comp. Classification
8. Summary

- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- D: Dissimilar
- Comp: Complementary

Empirical Risk Minimization Framework for Weakly Supervised Learning

- Any loss, classifier, regularizer, and optimizer can be used.



Towards More Reliable ML

- Reliability for expectable situations:
  - Model the corruption process explicitly and correct the solution.
    - How to handle modeling error?
- Reliability for unexpected situations:
  - Consider worst-case robustness (“min-max”).
    - How to make it less conservative?
  - Include human support (“rejection”).
    - How to handle real-time applications?
- Exploring somewhere in the middle would be practically more useful:
  - Use partial knowledge of the corruption process.