KBYÖYO2019 June 25, 2019

Machine Learning from Weak Supervision:

Towards Accurate Classification with Low Labeling Costs

Slides:
http://goo.gl/meiTwY

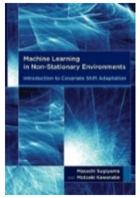
Masashi Sugiyama

RIKEN Center for Advanced Intelligence Project/
The University of Tokyo

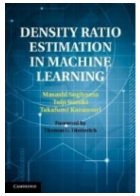
About Myself

- **Affiliations:**
  - Director: RIKEN AIP
  - Professor: University of Tokyo
  - Consultant: several local startups
- **Research interests:**
  - Theory and algorithms of ML
  - Real-world applications with partners (signal, image, language, brain, cars, robots, optics, ads, medicine, biology...)
- **Goal:**
  - Develop practically useful algorithms that have theoretical support

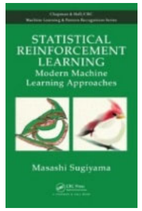
Sugiyama & Kawanabe, Machine Learning in Non-Stationary Environments, MIT Press, 2012



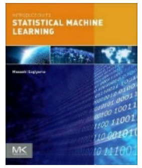
Sugiyama, Suzuki & Kanamori, Density Ratio Estimation in Machine Learning, Cambridge University Press, 2012



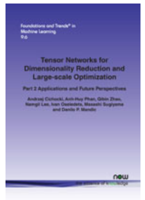
Sugiyama, Statistical Reinforcement Learning, Chapman and Hall/CRC, 2015



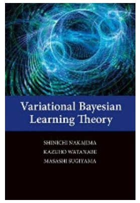
Sugiyama, Introduction to Statistical Machine Learning, Morgan Kaufmann, 2015



Cichocki, Phan, Zhao, Lee, Oseledets, Sugiyama & Mandic, Tensor Networks for Dimensionality Reduction and Large-Scale Optimizations, Now, 2017



Nakajima, Watanabe & Sugiyama, Variational Bayesian Learning Theory, Cambridge University Press, 2019



What Is This Tutorial about?

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

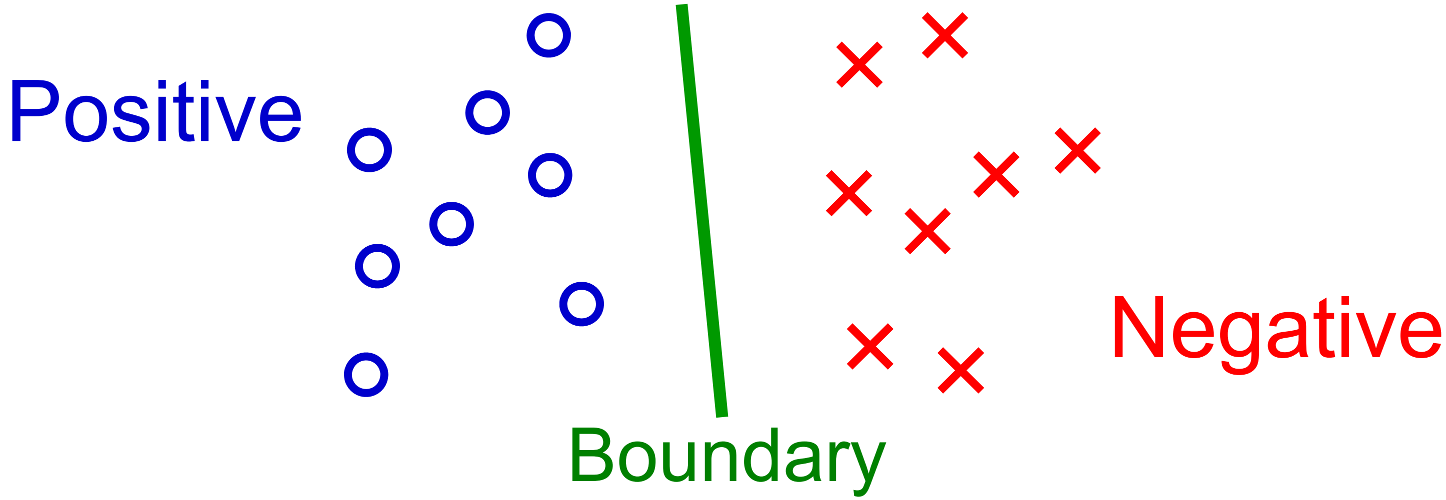
- However, there are various applications where massive labeled data is not available.
  - Medicine, disaster, robots, brain, …

What Is This Tutorial 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)

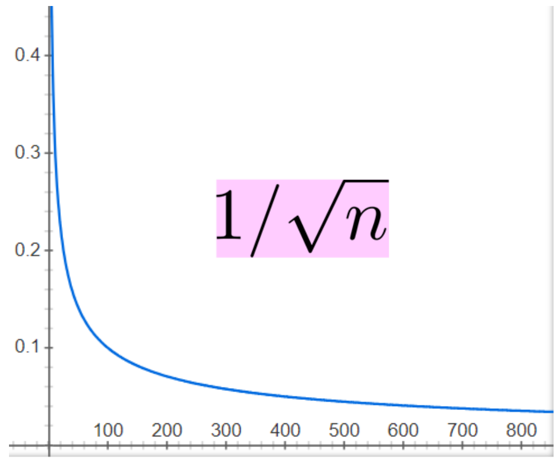
- I introduce our recent advances in classification from weak supervision.

Our Target Problem: Binary Supervised Classification



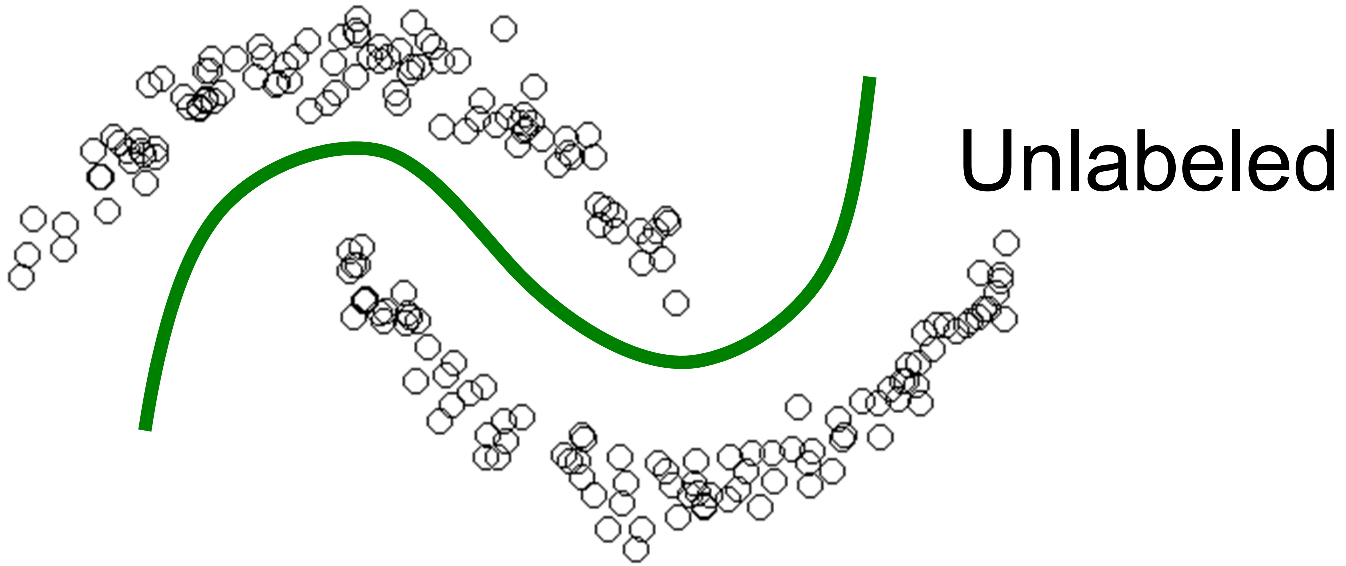
- Larger amount of labeled data yields better classification accuracy.
- Estimation error of the boundary decreases in order $1/\sqrt{n}$.

n : Number of labeled samples



Unsupervised Classification

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

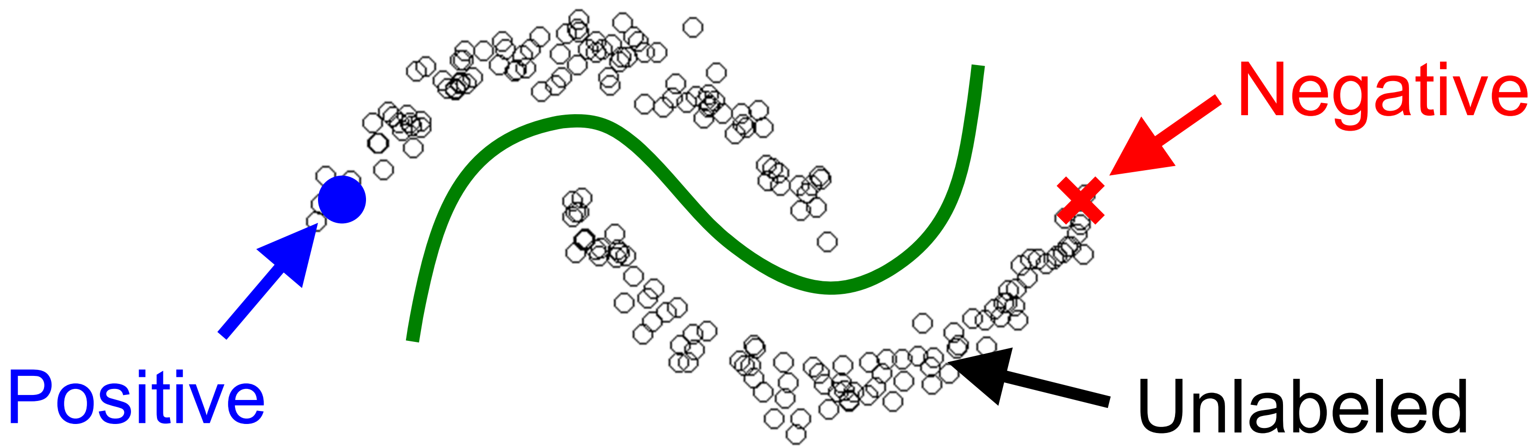


- Unsupervised classification is typically clustering.
- This works well only when each cluster corresponds to a class.

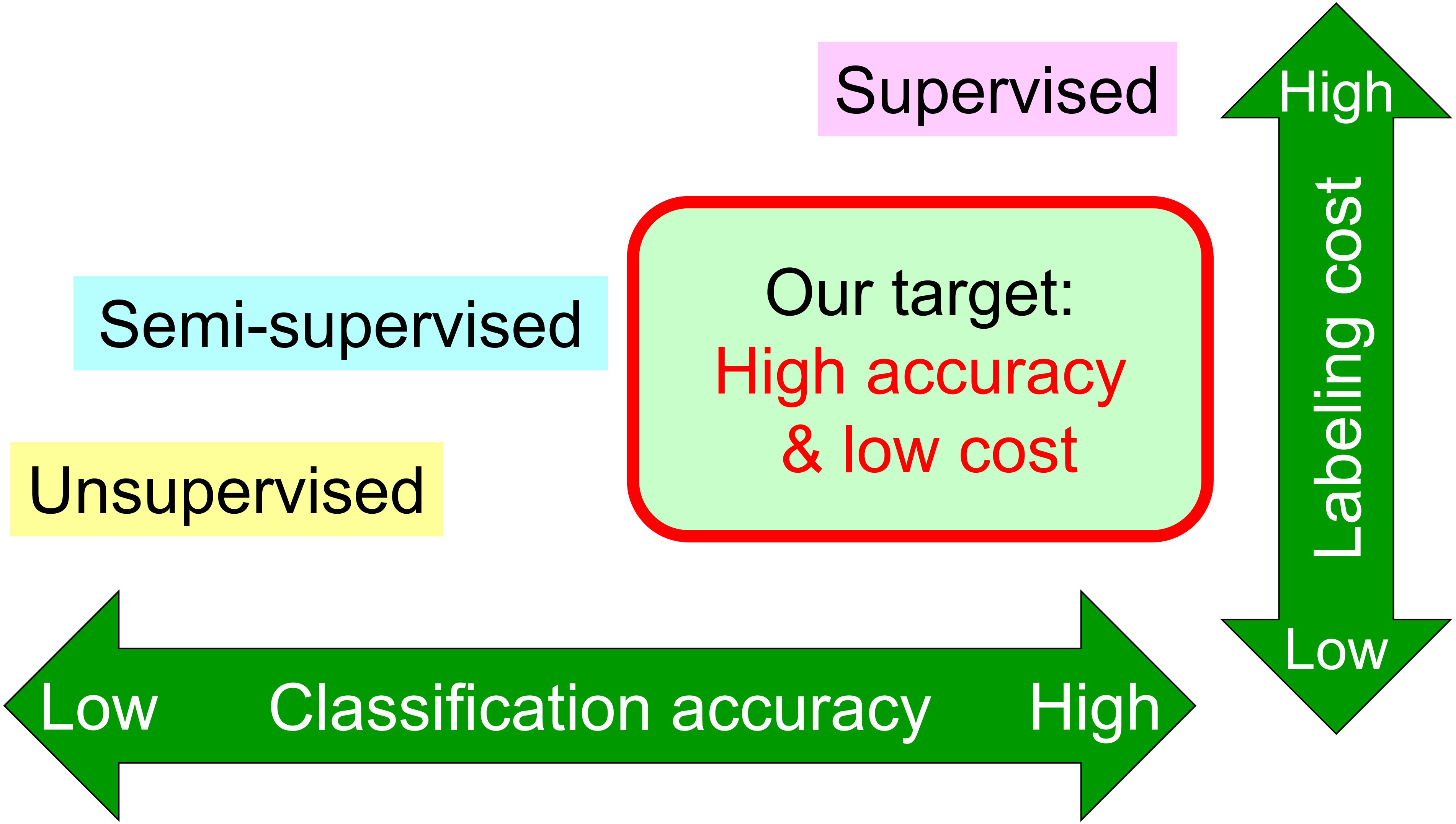
Semi-Supervised Classification

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



This Tutorial in a Nutshell

1. Background
2. PN Classification
3. PU Classification
4. PNU Classification
5. Pconf Classification
6. UU Classification
7. SU Classification
8. Comp Classification
9. Summary

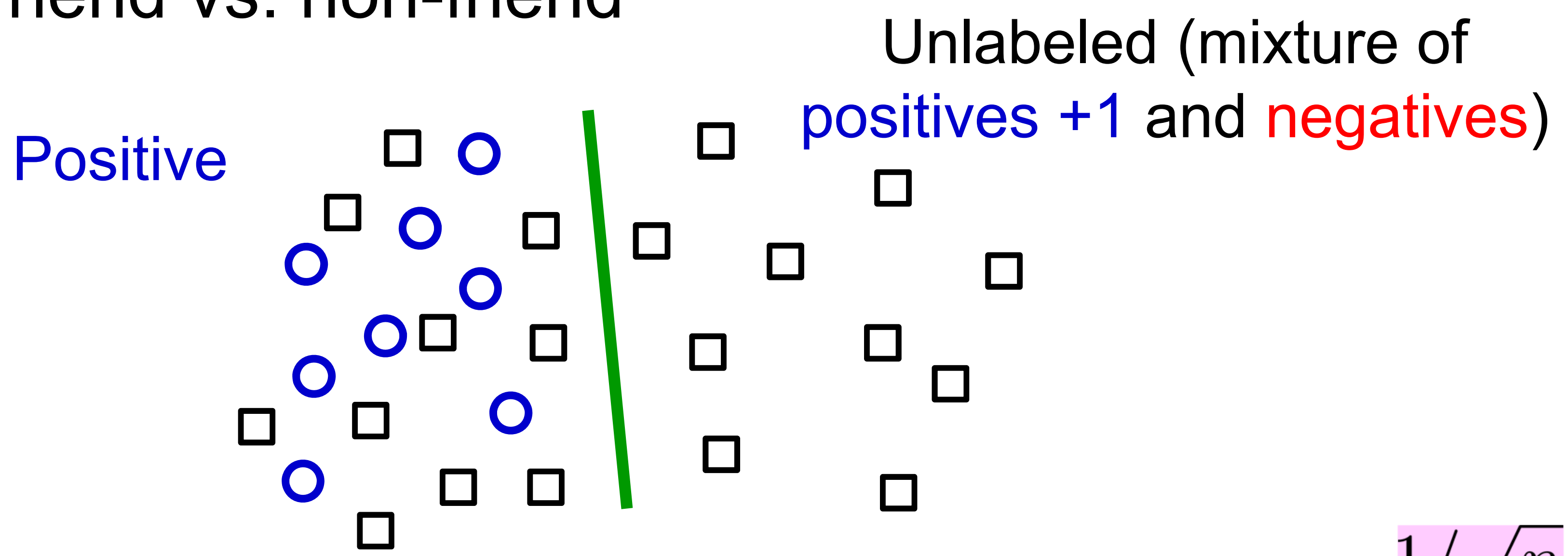
- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- Comp: Complementary

Slides:
http://goo.gl/meiTwY

Method 1: 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



$1/\sqrt{n}$

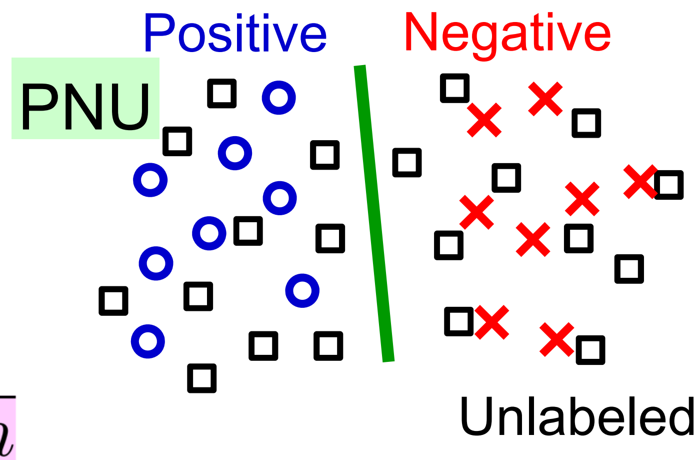
- From PU data, PN classifiers are trainable!

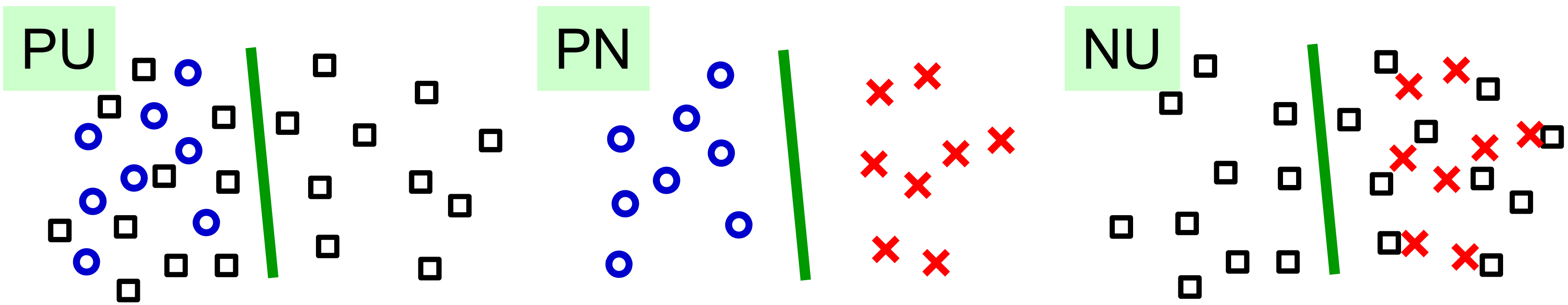
Method 2: PNU Classification (Semi-Supervised Classification)

Sakai, du Plessis, Niu & Sugiyama (ICML2017)

- Let’s decompose PNU into PU, PN, and NU:
  - Each is solvable.
  - Let’s combine them!
- Without cluster assumptions, PN classifiers are trainable!

$1/\sqrt{n}$





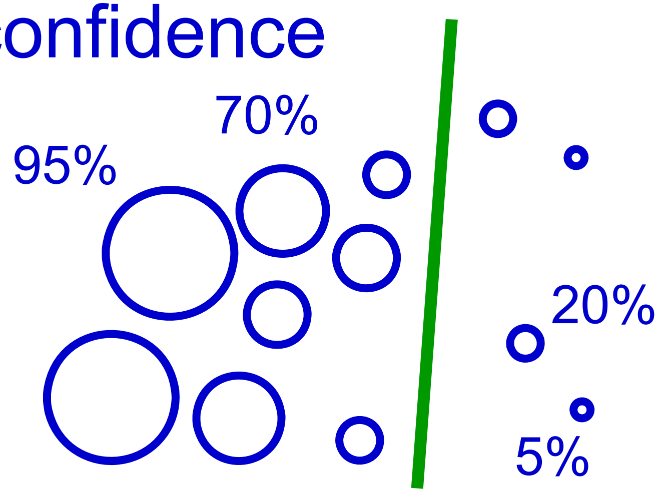
Method 3: Pconf Classification

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!

$1/\sqrt{n}$

Positive confidence

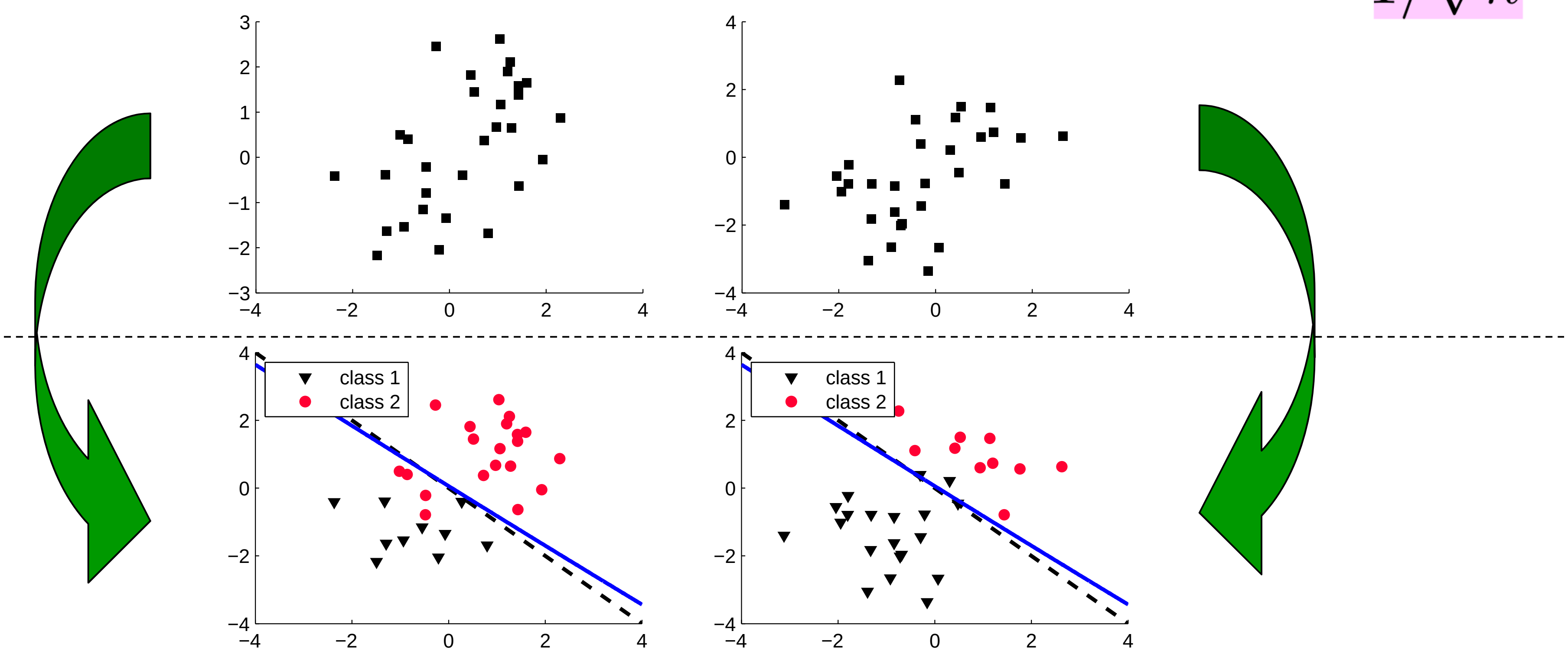


Method 4: UU Classification

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

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

$1/\sqrt{n}$

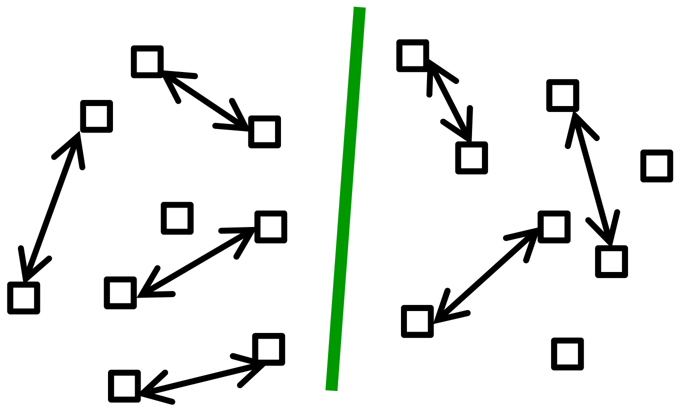


Method 5: SU Classification

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!

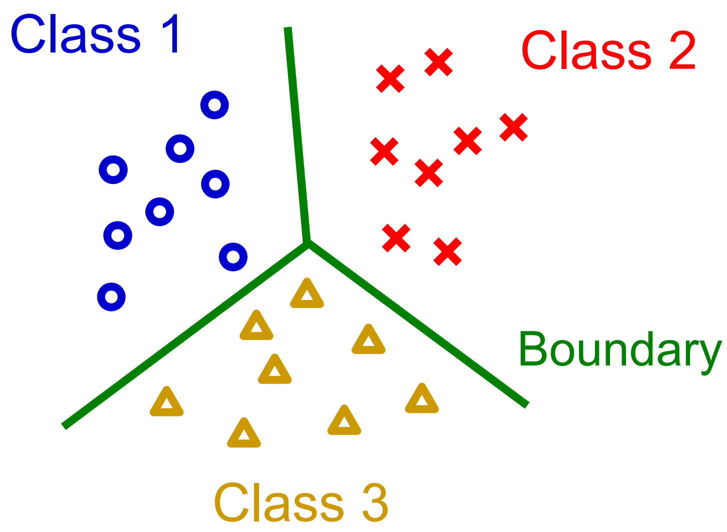
$1/\sqrt{n}$



Method 6: Comp Classification

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

- Labeling patterns in multi-class problems:
  - Selecting a collect 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!



$1/\sqrt{n}$

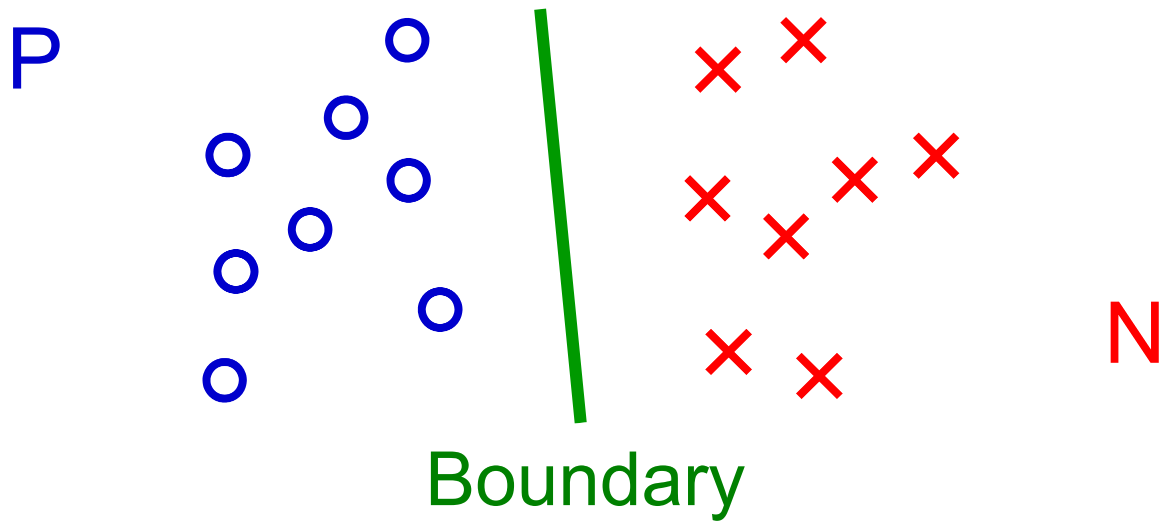
Contents

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

Slides:
http://goo.gl/meiTwY

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**: $m = yf(\boldsymbol{x})$ $y \in \{+1, -1\}$
  - $m > 0 \implies \mathrm{sign}(f(\boldsymbol{x})) = y$
    → Classification is correct.
  - $m < 0 \implies \mathrm{sign}(f(\boldsymbol{x})) \neq y$
    → Classification is wrong.
- **Zero-one loss**: $\ell_{0/1}(m) = \dfrac{1}{2}\left(1 - \mathrm{sign}\left(m\right)\right)$
  - 1 for correct prediction.
  - 0 for wrong prediction.

Classification Error and Empirical Approximation

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

$\mathbb{E}$: Expectation

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

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

  - 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:

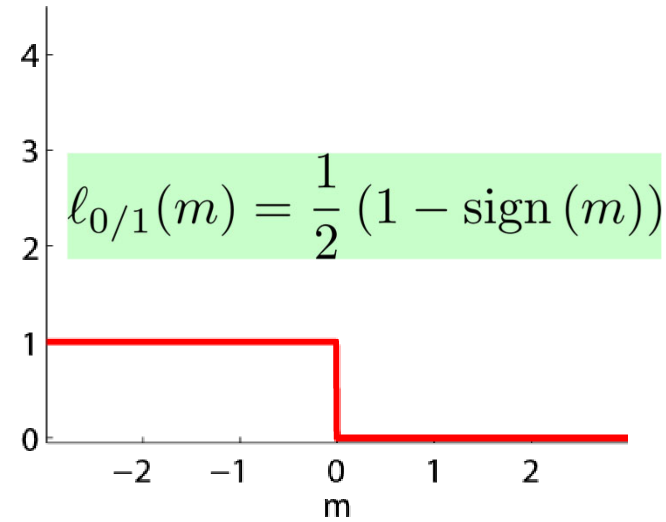
$$\widehat{R}_{0/1}(f) = \frac{1}{n}\sum_{i=1}^{n}\ell_{0/1}\Big(y_i f(\boldsymbol{x}_i)\Big) = R_{0/1}(f) + O_p\left(\frac{1}{\sqrt{n}}\right)$$

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.

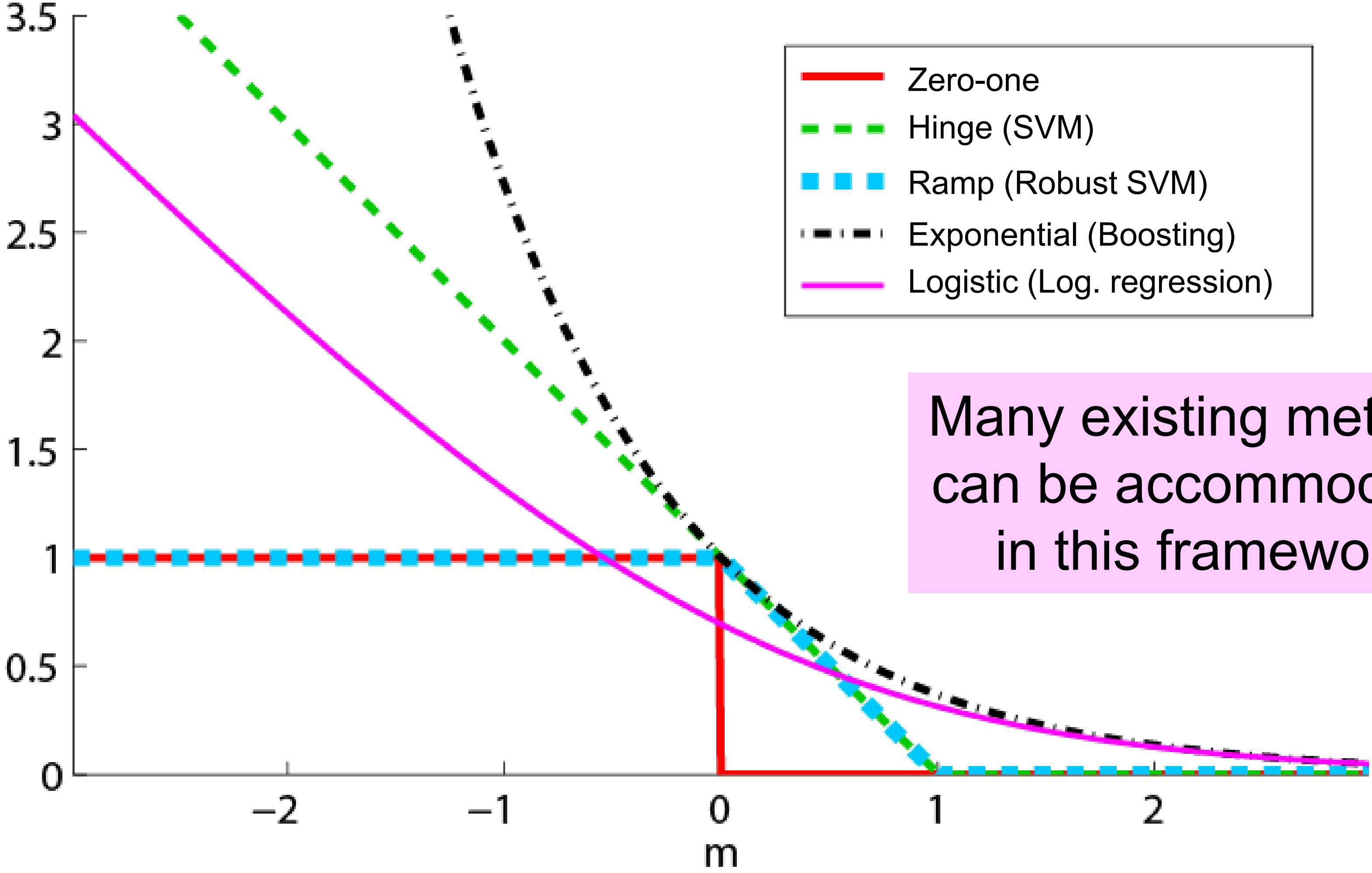
- Let's use a smoother loss!



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

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 :

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

- 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,…):

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

Contents

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

Slides:
http://goo.gl/meiTwY

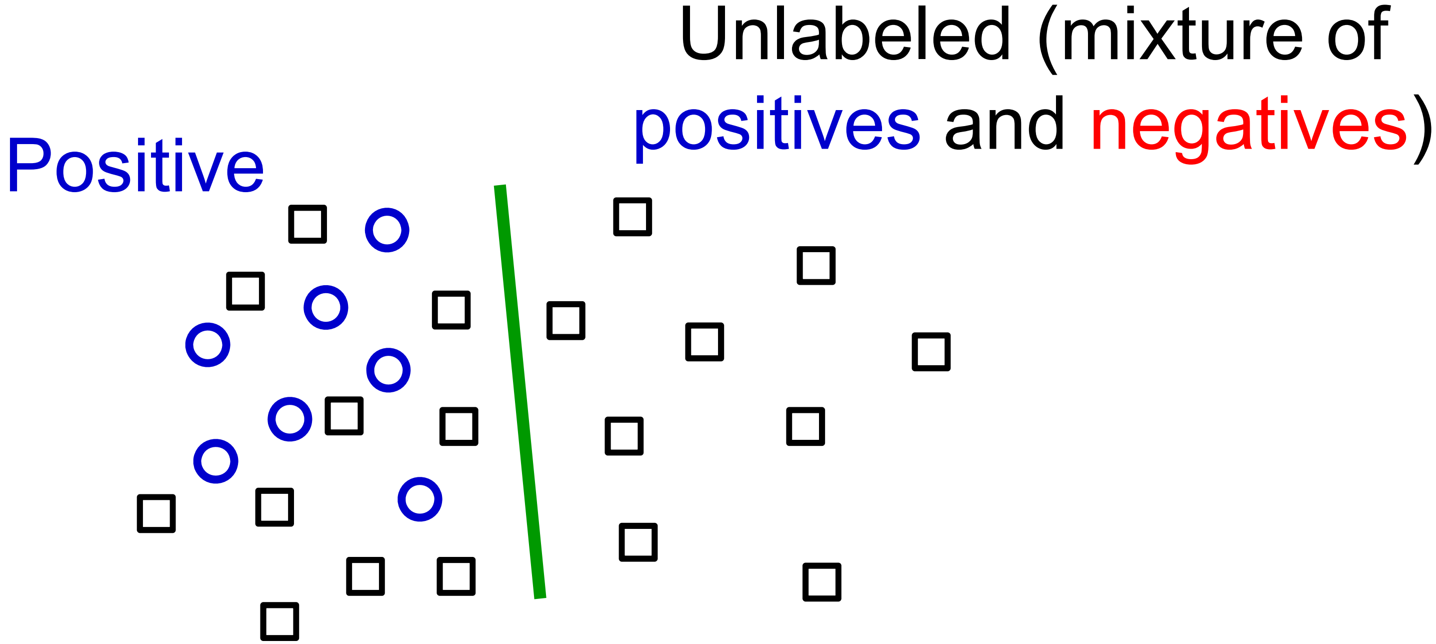
PU Classification: Setup

- Given: Positive and unlabeled samples

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

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

- Goal: Obtain a PN classifier



PN Risk Decomposition

- Risk of classifier f :

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

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

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

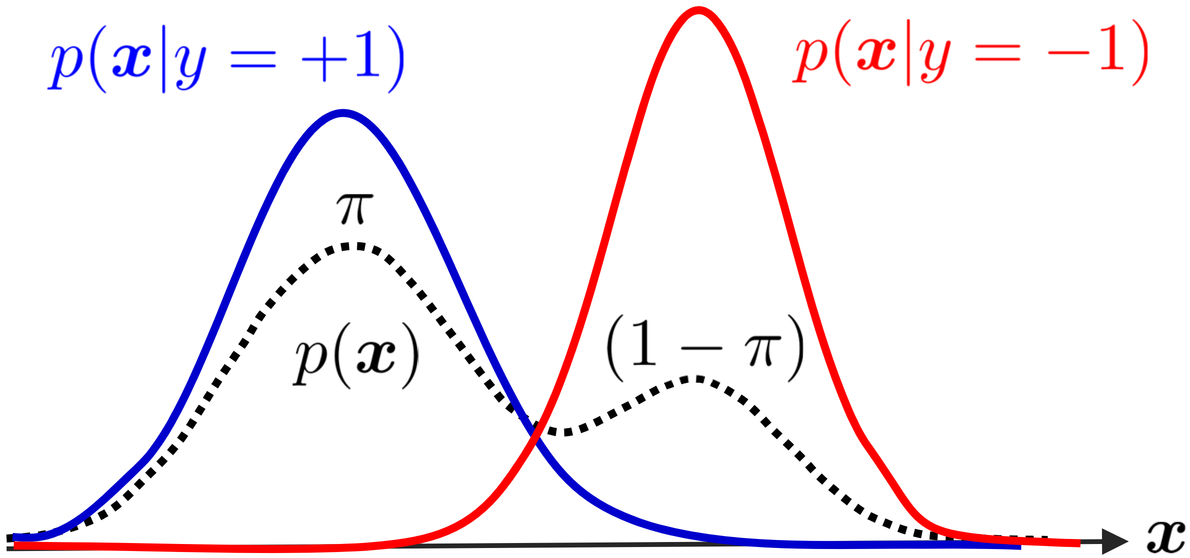
- Since we do not have N data in the PU setting, the risk cannot be directly estimated.

PU Risk Estimation

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

- 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

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

- Replacing expectations by sample averages gives an empirical risk:

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

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

- 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

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

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

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

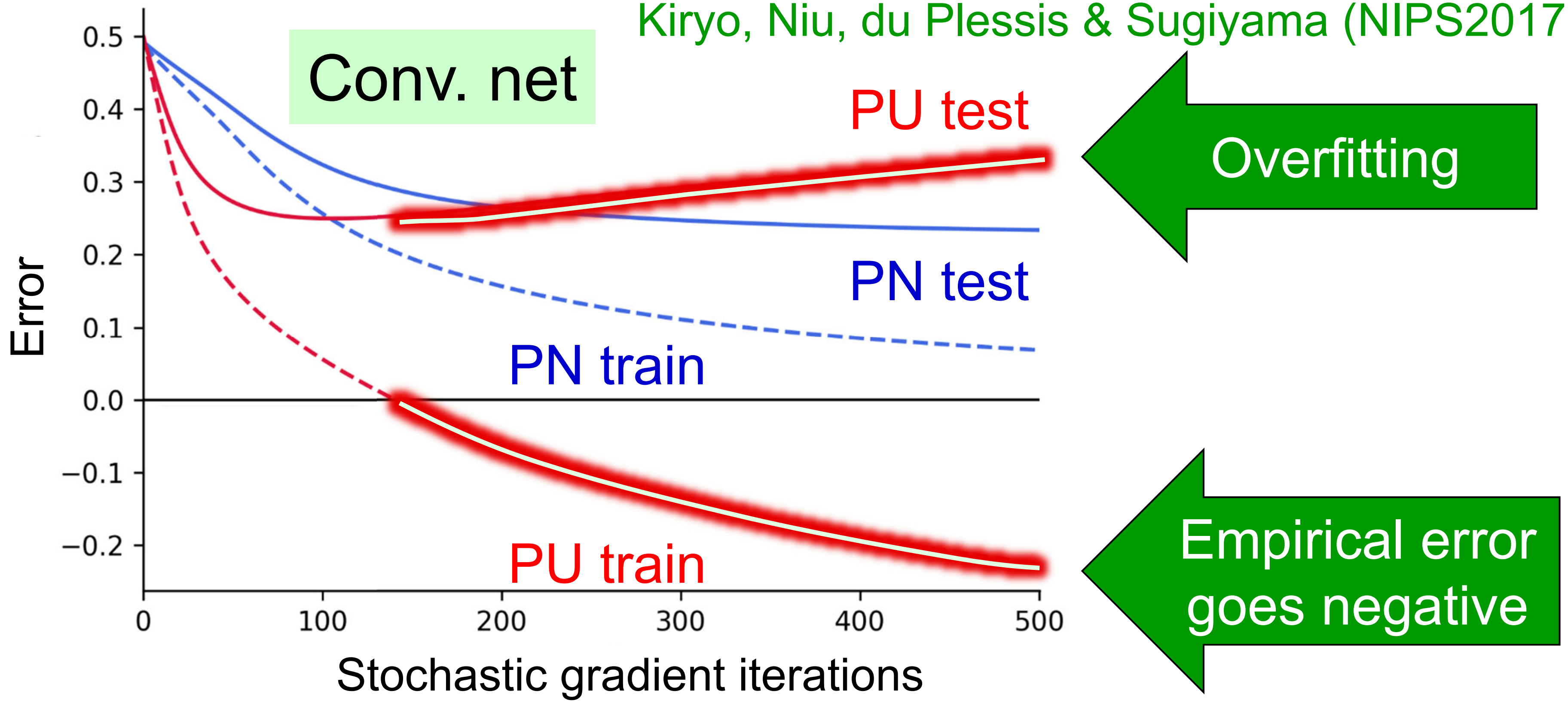
  - If $\ell(m) \geq 0, \ \forall m$, $R^{-}(f) \geq 0$.
  - However, its PU empirical approximation can be negative due to “difference of approximations”.

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

  - 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

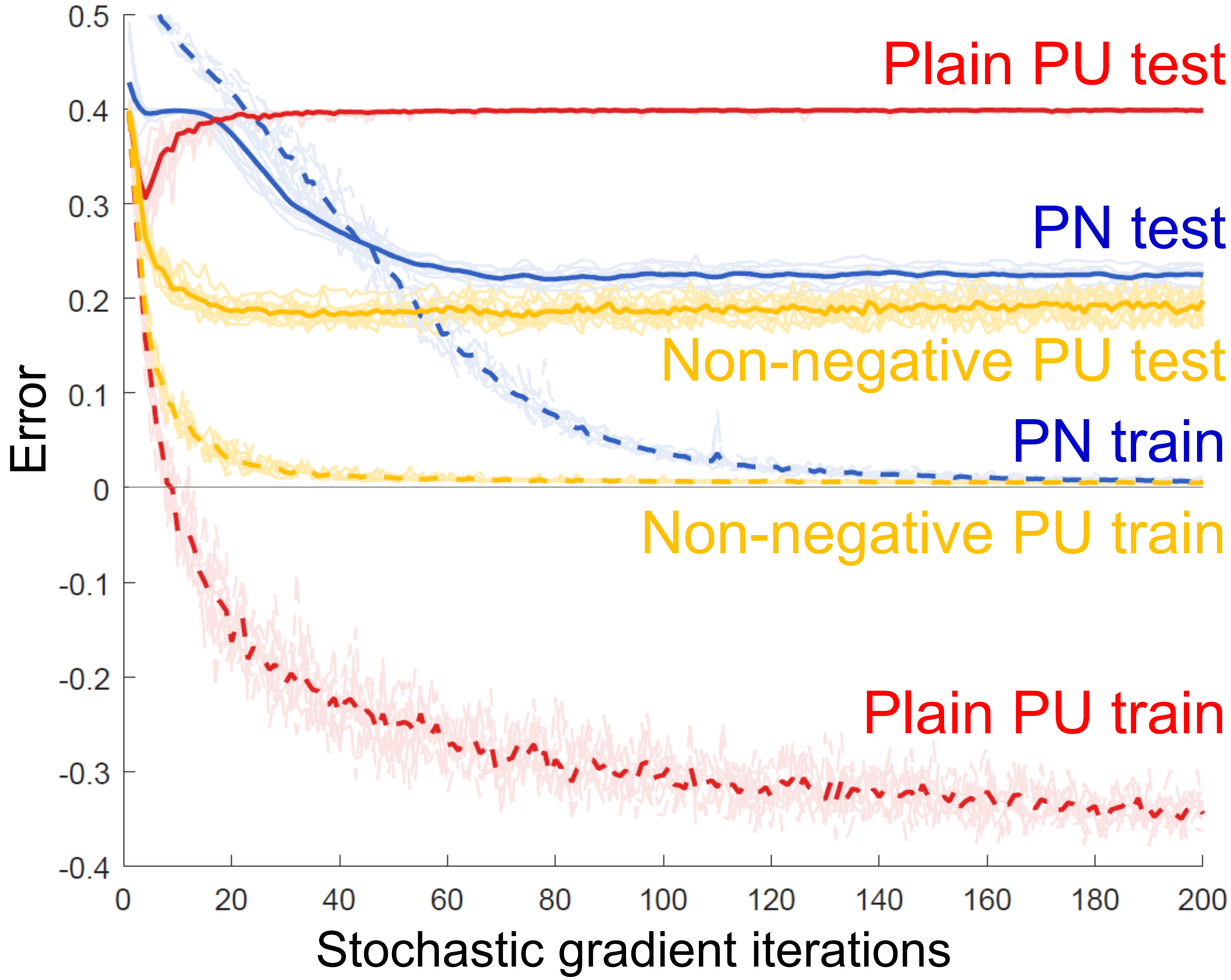
  - In practice, we can ignore the bias of $\widetilde{R}_{\mathrm{PU}}(f)$!
- Mean-squared error of $\widetilde{R}_{\mathrm{PU}}(f)$ is not more than the original one.
  - In practice, $\widetilde{R}_{\mathrm{PU}}(f)$ is more reliable!
- Risk of $\mathrm{argmin}_f\, \widetilde{R}_{\mathrm{PU}}(f)$ for 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.

Experiments

- With a large number of unlabeled data, non-negative PU can even outperform PN!

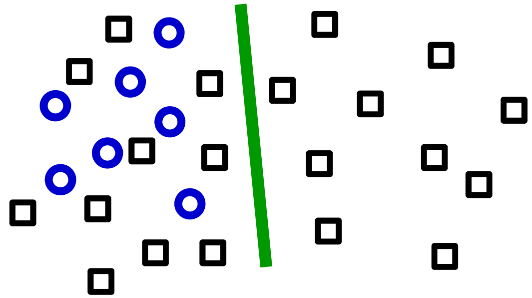
- Binary CIFAR-10: Positive (airplane, automobile, ship, truck) Negative (bird, cat, deer, dog, frog, horse)
- 13-layer CNN with ReLU

$n_{\mathrm{P}} = 1000$
$n_{\mathrm{U}} = 50000$
$\pi = 0.4$



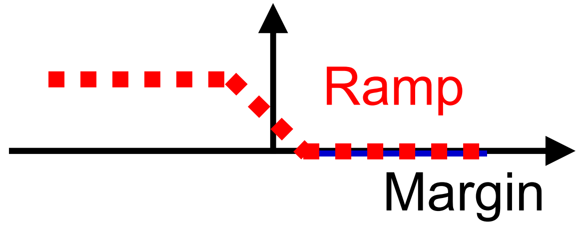
PU Classification: Summary

- Just separating P and U is biased.



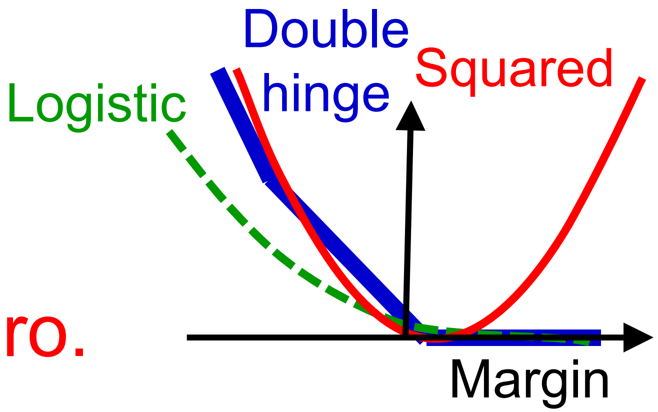
- To be unbiased, use composite loss $\widetilde{\ell}(m) = \ell(m) - \ell(-m)$ for P data.
  Natarajan, Dhillon, Ravikumar & Tewari (NIPS2013)
  - Optimal convergence rate achieved.
    Niu, du Plessis, Sakai, Ma & Sugiyama (NIPS2016)

$\ell(m) + \ell(-m) = \mathrm{Const.}$



- If $\ell(m) + \ell(-m) = \mathrm{Const.}$,
  the same loss for P and U data.
  du Plessis, Niu & Sugiyama (NIPS2014)

$\widetilde{\ell}(m) = am + b$



- If $\widetilde{\ell}(m) = am + b$,
  optimization becomes convex.
  du Plessis, Niu & Sugiyama (ICML2015)
- For deep nets, roundup the empirical false negative error to zero.
  Kiryo, Niu, du Plessis & Sugiyama (NIPS2017)

Contents

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

Slides:
http://goo.gl/meiTwY

PNU Classification: Setup

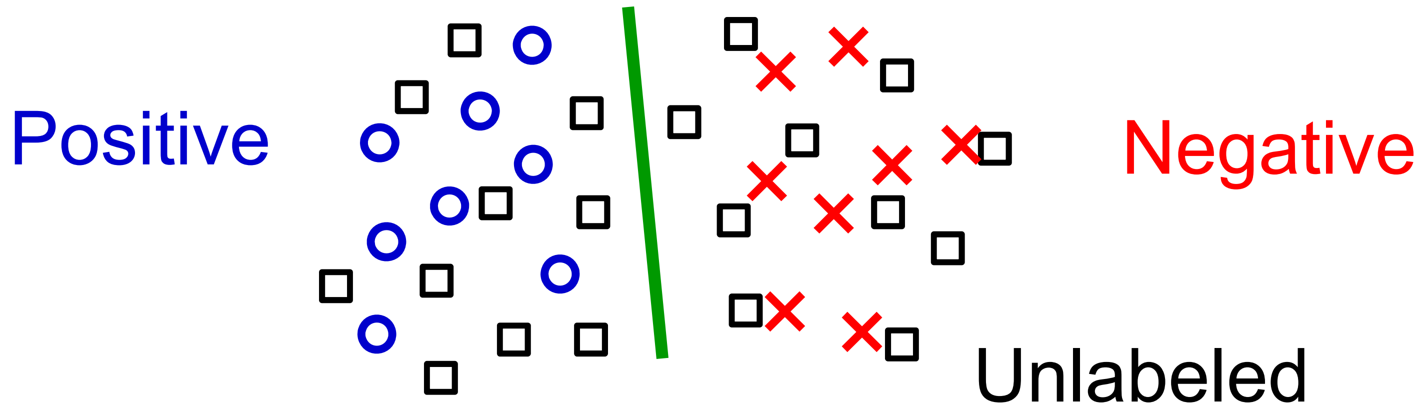
- Given: Positive, negative & unlabeled samples

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

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

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

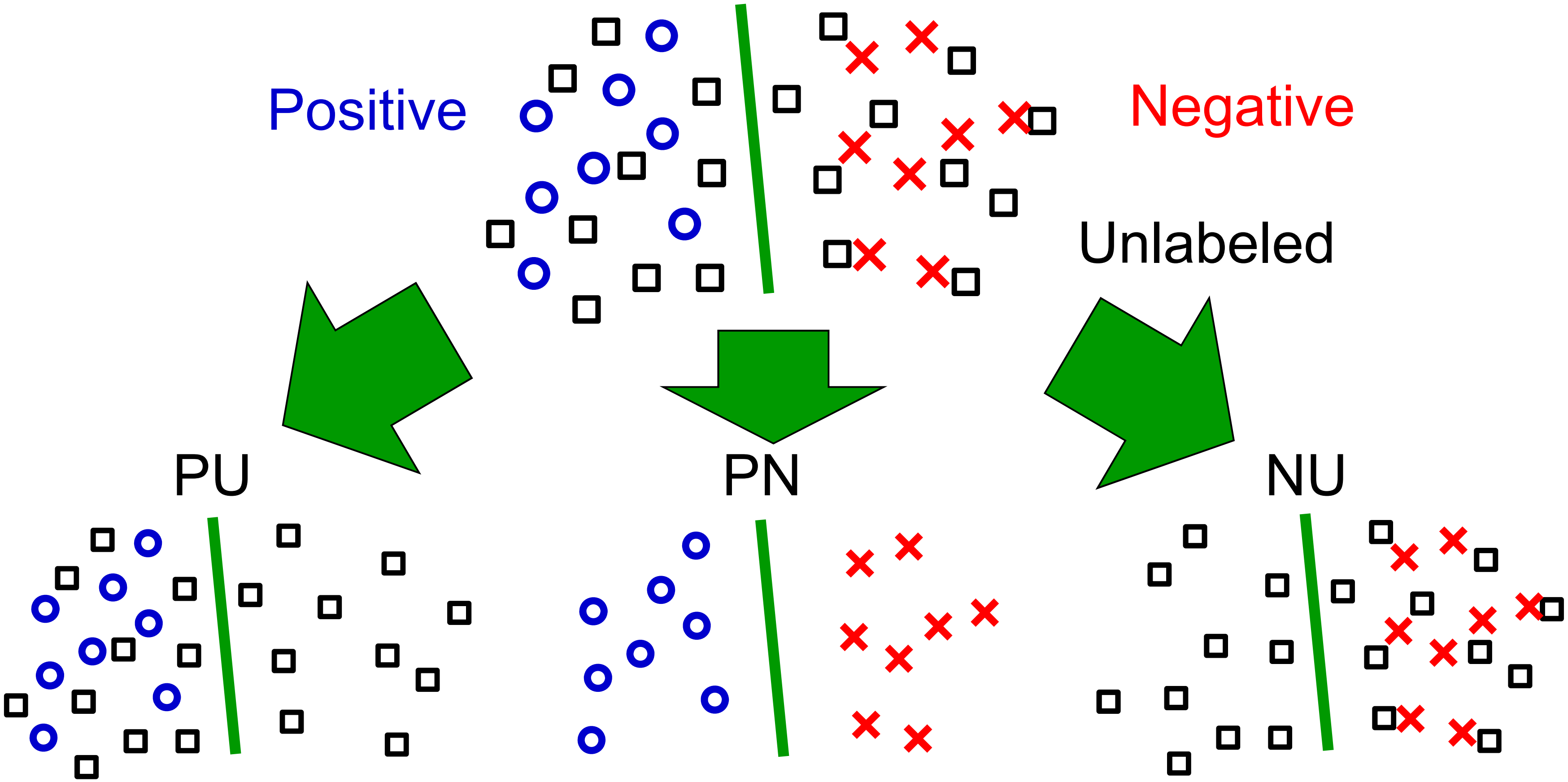
- Goal: Obtain a PN classifier
  - PNU classification is semi-supervised learning.



PNU Decomposition

Sakai, du Plessis, Niu & Sugiyama (ICML2017)

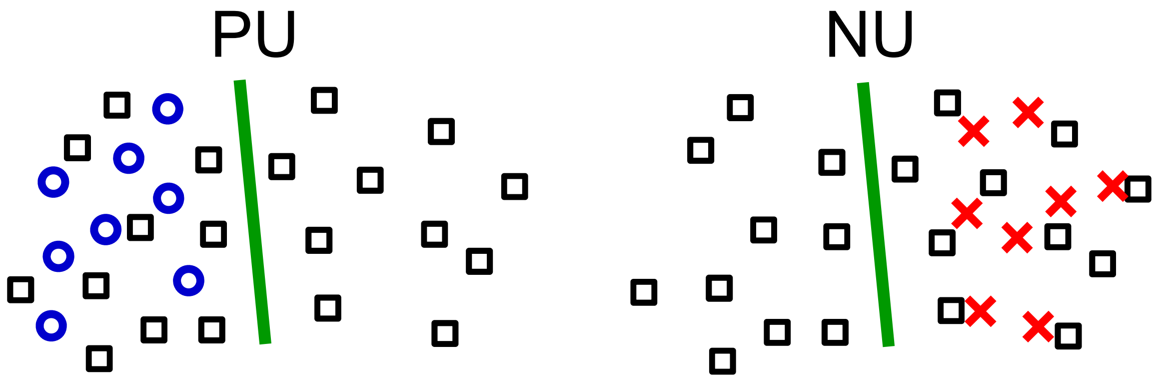
- Let's decompose PNU into PU, PN, and NU:
  - Each can be solved easily.
  - Combine them!



How to Combine?

- Natural choice: Combine PU & NU (symmetric).

$$R_{\mathrm{PU+NU}}(f) = (1-\gamma)R_{\mathrm{PU}}(f) + \gamma R_{\mathrm{NU}}(f) \qquad 0 \leq \gamma \leq 1$$



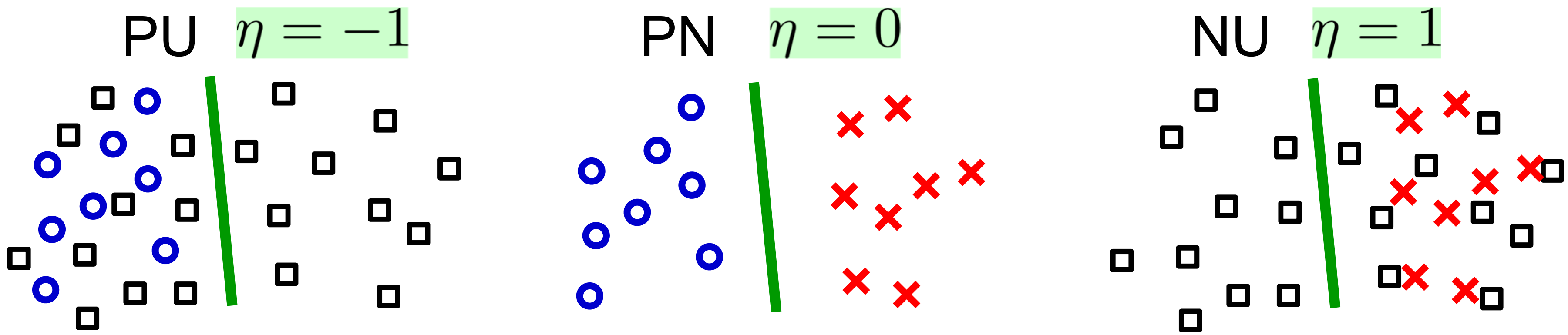
- Theoretical risk analysis:

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

  - When PU<NU: PU<PN<NU or PN<PU<NU.
  - When NU<PU: NU<PN<PU or PN<NU<PU.
- PU+NU is not the best possible combination.
- PU+PN & NU+PN are the best combinations.

PN+PU & PN+NU Classification

- **Proposed method:** Combine two best methods.



$$R^{\eta}_{\mathrm{PNU}}(f) = \begin{cases} R^{\eta}_{\mathrm{PN+PU}}(f) & (\eta \geq 0) \\ R^{-\eta}_{\mathrm{PN+NU}}(f) & (\eta < 0) \end{cases} \qquad -1 \leq \eta \leq 1$$

- PN+PU classification:

$$R^{\gamma}_{\mathrm{PN+PU}}(f) = (1-\gamma)R_{\mathrm{PN}}(f) + \gamma R_{\mathrm{PU}}(f) \qquad 0 \leq \gamma \leq 1$$

- PN+NU classification:

$$R^{\gamma}_{\mathrm{PN+NU}}(f) = (1-\gamma)R_{\mathrm{PN}}(f) + \gamma R_{\mathrm{NU}}(f) \qquad 0 \leq \gamma \leq 1$$

Theoretical Analysis

- Generalization error bound:

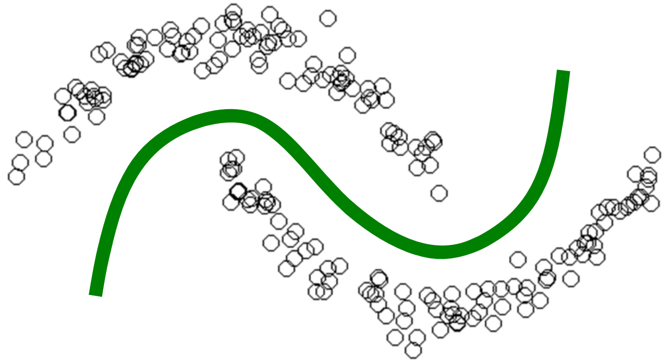
$$R_{0/1}(f) \leq 2\widehat{R}^{\eta}_{\mathrm{PNU}}(f) + \mathcal{O}\left(\frac{1}{\sqrt{n_{\mathrm{P}}}} + \frac{1}{\sqrt{n_{\mathrm{N}}}} + \boxed{\frac{1}{\sqrt{n_{\mathrm{U}}}}}\right)$$

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

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

$\widehat{R}^{\eta}_{\mathrm{PNU}}(f)$: Empirical version of $R^{\eta}_{\mathrm{PNU}}(f) = \begin{cases} R^{\eta}_{\mathrm{PN+PU}}(f) & (\eta \geq 0) \\ R^{-\eta}_{\mathrm{PN+NU}}(f) & (\eta < 0) \end{cases}$

  - Unlabeled data always helps without cluster assumptions!



- We use unlabeled data for loss evaluation, not for regularization (as manifold smoothing).
  - Label information is extracted from unlabeled data!

Experiments

Misclassification error rate: average (std)

5% t-test

Dataset	n_u	π	$\widehat{\pi}$	Proposed	EntReg (Grandvalet & Bengio, NIPS2004)	LapSVM (Belkin et al., JMLR2006)	SMIR (Niu et al., ICML2013)	WellSVM (Li et al., JMLR2013)
Arts	1000	0.50	0.49 (0.01)	27.4 (1.3)	26.6 (0.5)	26.1 (0.7)	40.1 (3.9)	27.5 (0.5)
	5000	0.50	0.50 (0.01)	24.8 (0.6)	26.1 (0.5)	26.1 (0.4)	30.1 (1.6)	N/A
	10000	0.50	0.52 (0.01)	25.6 (0.7)	25.4 (0.5)	25.5 (0.6)	N/A	N/A
Deserts	1000	0.73	0.67 (0.01)	13.0 (0.5)	15.3 (0.6)	16.7 (0.8)	17.2 (0.8)	18.2 (0.7)
	5000	0.73	0.67 (0.01)	13.4 (0.4)	13.3 (0.5)	16.6 (0.6)	24.4 (0.6)	N/A
	10000	0.73	0.68 (0.01)	13.3 (0.5)	13.7 (0.6)	16.8 (0.8)	N/A	N/A
Fields	1000	0.65	0.57 (0.01)	22.4 (1.0)	26.2 (1.0)	26.6 (1.3)	28.2 (1.1)	26.6 (0.8)
	5000	0.65	0.57 (0.01)	20.6 (0.5)	22.6 (0.6)	24.7 (0.8)	29.6 (1.2)	N/A
	10000	0.65	0.57 (0.01)	21.6 (0.6)	22.5 (0.6)	25.0 (0.9)	N/A	N/A
Stadiums	1000	0.50	0.50 (0.01)	11.4 (0.4)	11.5 (0.5)	12.5 (0.5)	17.4 (3.6)	11.7 (0.4)
	5000	0.50	0.50 (0.01)	11.0 (0.5)	10.9 (0.3)	11.1 (0.3)	13.4 (0.7)	N/A
	10000	0.50	0.51 (0.00)	10.7 (0.3)	10.9 (0.3)	11.2 (0.2)	N/A	N/A
Platforms	1000	0.27	0.33 (0.01)	21.8 (0.5)	23.9 (0.6)	24.1 (0.5)	30.1 (2.3)	26.2 (0.8)
	5000	0.27	0.34 (0.01)	23.3 (0.8)	24.4 (0.7)	24.9 (0.7)	26.6 (0.3)	N/A
	10000	0.27	0.34 (0.01)	21.4 (0.5)	24.3 (0.6)	24.8 (0.5)	N/A	N/A
Temples	1000	0.55	0.51 (0.01)	43.9 (0.7)	43.9 (0.6)	43.4 (0.6)	50.7 (1.6)	44.3 (0.5)
	5000	0.55	0.54 (0.01)	43.4 (0.9)	43.0 (0.6)	43.1 (1.0)	43.6 (0.7)	N/A
	10000	0.55	0.50 (0.01)	45.2 (0.8)	44.4 (0.8)	44.2 (0.7)	N/A	N/A

- Proposed PN+PU & PN+NU works well!

Contents

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

Slides:
http://goo.gl/meiTwY

Pconf Classification: Setup

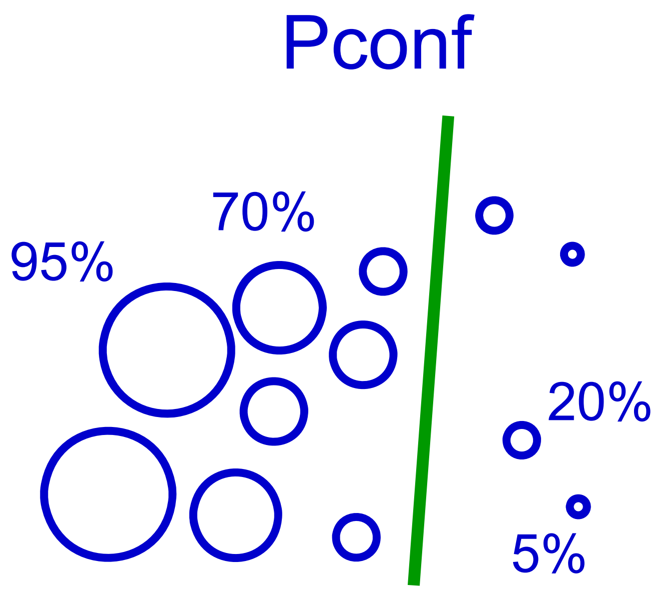
Ishida, Niu & Sugiyama (NeurIPS2018)

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

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

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

- Right form is given by importance sampling:

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

resulting in an empirical risk:

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

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

Theoretical Analysis

- Estimation error:

$$R(f^*) - R(\widehat{f}_{\mathrm{Pconf}}) = \mathcal{O}_p\left(\frac{1}{\sqrt{n}}\right)$$

$$f^* = \operatorname*{argmin}_f R(f) \qquad \widehat{f}_{\mathrm{Pconf}} = \operatorname*{argmin}_f \widehat{R}_{\mathrm{Pconf}}(f)$$

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

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

- Optimal parametric convergence rate is attained!

Experiments

Correct classification rate: average (std)

5% t-test

Positive	vs.	Negative	Pconf	Weighted	Supervised
airplane	vs.	automobile	**84.34 ± 0.84**	79.32 ± 2.74	93.82 ± 0.21
airplane	vs.	bird	**82.50 ± 3.19**	**81.38 ± 0.48**	89.24 ± 0.50
airplane	vs.	cat	**89.10 ± 0.47**	86.98 ± 1.20	92.78 ± 0.49
airplane	vs.	deer	**87.44 ± 1.43**	82.00 ± 2.39	92.08 ± 0.50
airplane	vs.	dog	**90.24 ± 1.27**	86.86 ± 1.41	94.42 ± 0.89
airplane	vs.	frog	**91.44 ± 0.86**	85.12 ± 1.66	95.52 ± 0.42
airplane	vs.	horse	**89.26 ± 2.20**	**87.72 ± 1.99**	95.58 ± 0.56
airplane	vs.	ship	**74.36 ± 2.00**	70.82 ± 1.80	89.04 ± 1.06
airplane	vs.	truck	**84.98 ± 0.47**	83.22 ± 0.58	91.84 ± 1.19

- Works better than naïve “weighted” baseline!

Contents

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

Slides:
http://goo.gl/meiTwY

UU Classification: Setup

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

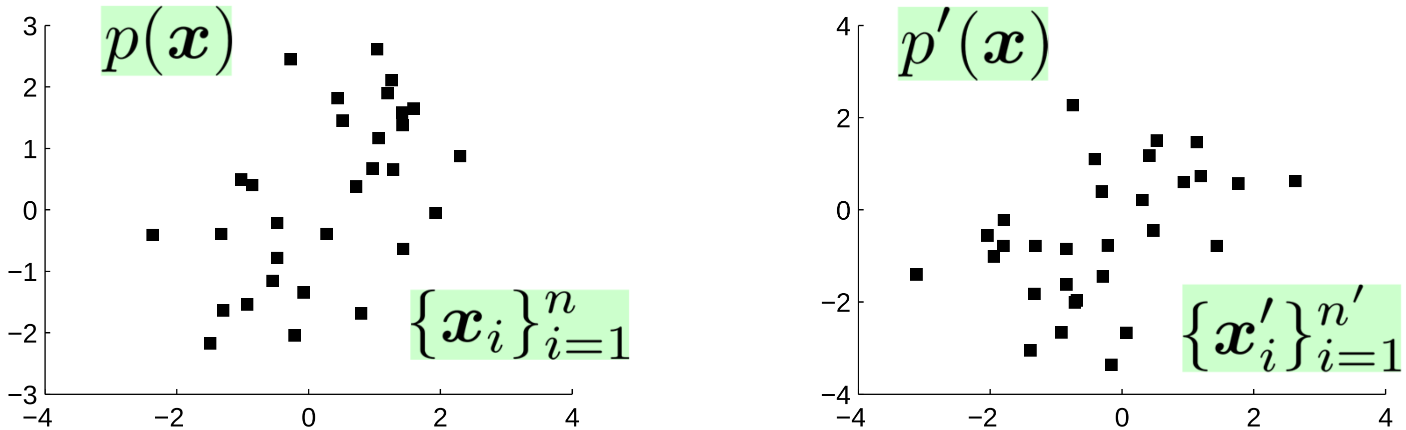
- Given: Two sets of unlabeled data

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

- 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,

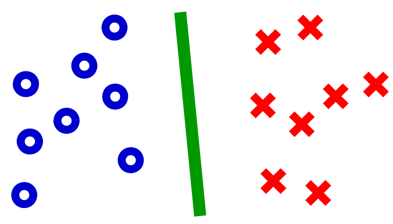
$$g(\boldsymbol{x}) = C\mathrm{sign}[p(\boldsymbol{x}) - p'(\boldsymbol{x})]$$

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

- Sign of C is unknown, but just knowing

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

still allows optimal separation!



UU Risk Estimation

Nan, Niu, Menon & Sugiyama (ICLR2019)

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

  the classification risk can be expressed as

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

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

  resulting an empirical risk (up to label flip):

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

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

Theoretical Analysis

- Estimation error:

$$R(f^*) - R(\widehat{f}_{\mathrm{UU}}) = \mathcal{O}_p\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{n'}}\right)$$

$$f^* = \mathop{\mathrm{argmin}}_f R(f) \qquad \widehat{f}_{\mathrm{UU}} = \mathop{\mathrm{argmin}}_f \widehat{R}_{\mathrm{UU}}(f)$$

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

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

- Optimal parametric convergence rate is attained!

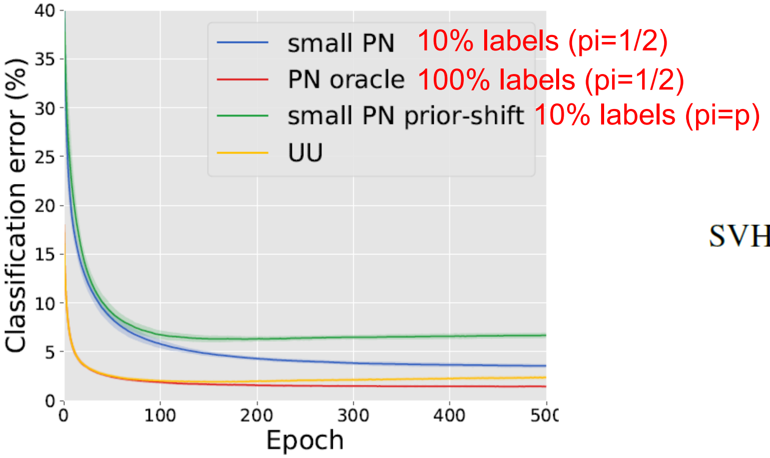
Experiments

Dataset	# Train	# Test	# Feature	π_P	Model $g(x;\theta)$	Optimizer
MNIST	60,000	10,000	784	0.49	FC with ReLU (depth 5)	SGD
Fashion-MNIST	60,000	10,000	784	0.50	FC with ReLU (depth 5)	SGD
SVHN	100,000	26,032	3,072	0.27	AllConvNet (depth 12)	Adam
CIFAR-10	50,000	10,000	3,072	0.60	ResNet (depth 32)	Adam

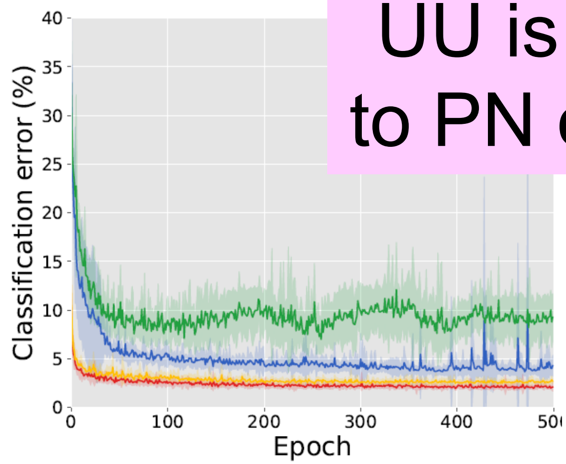
$p(y=+1)=0.9,\ p'(y=+1)=0.1$

UU is close to PN oracle!

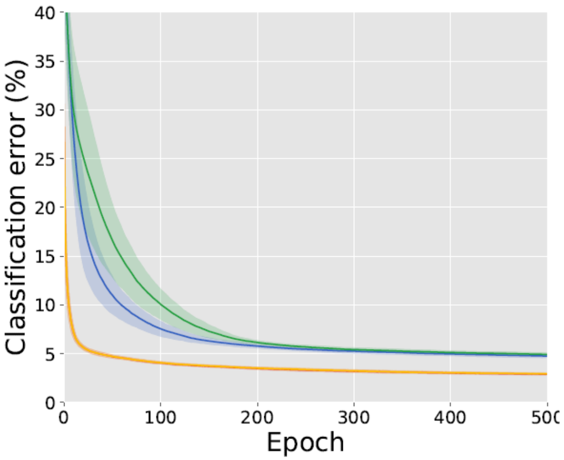
MNIST



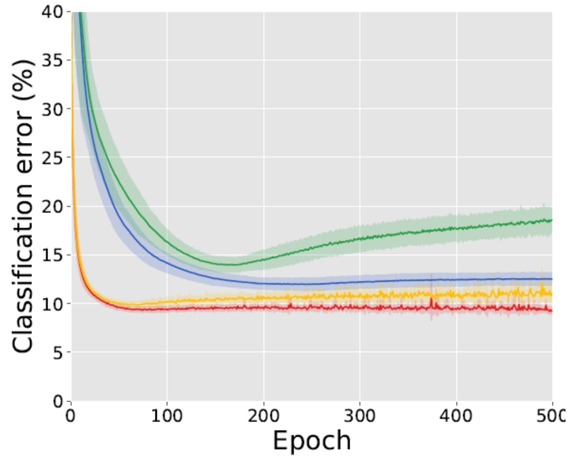
SVHN



Fashion-MNIST



CIFAR-10



Contents

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

Slides:
http://goo.gl/meiTwY

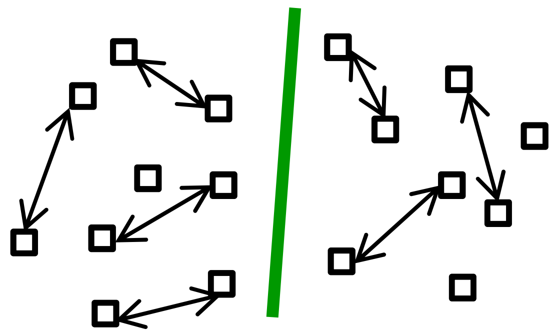
SU Classification

Bao, Niu & Sugiyama (ICML2018)

- Given: Similar and unlabeled samples

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

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

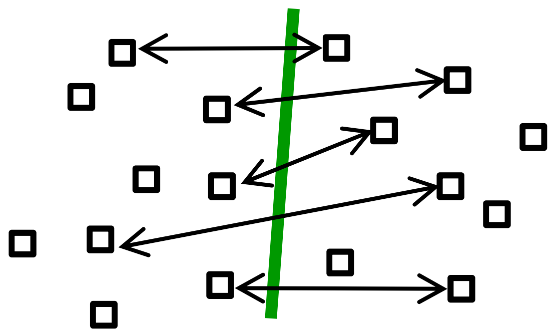


- Goal: Obtain a PN classifier
- This is a special case of UU classification:

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

- Classification from dissimilar data is also possible (DU, SD, SDU)!

Shimada, Bao, Sato & Sugiyama (arXiv2019)



Contents

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

Slides:
http://goo.gl/meiTwY

Multiclass Labeling is Costly

- Labeling in multi-class classification:
  - What is the robot in this image?

https://www.bostondynamics.com/atla

1. Amazon Kiva
2. Aldebaran Nao
3. Softbank Pepper
4. Sony Aibo
5. iRobot Roomba

⋮

83. Boston Dynamics Atlas

⋮

100. Rethink Robotics Baxter

- Selecting the correct class from a long list of candidates is extremely time-consuming!

Complementary Classification

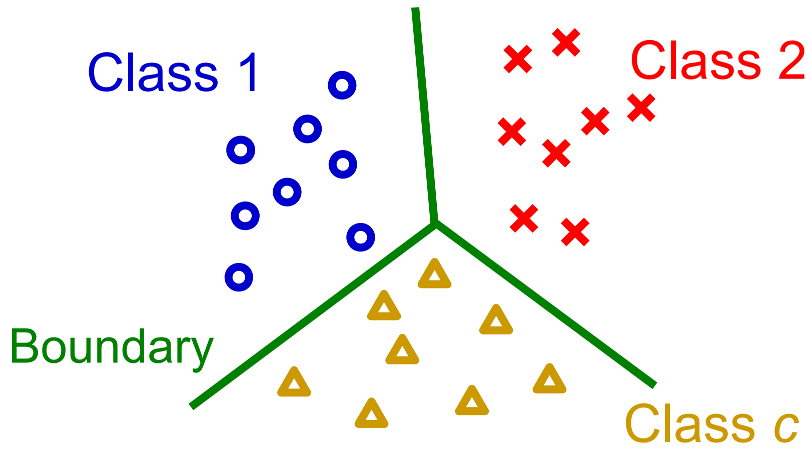
Ishida, Niu, Hu & Sugiyama (NIPS2017)
Ishida, Niu, Menon & Sugiyama (ICML2019)

- **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



Possible Approaches

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

- Approach 1: Classification from partial labels

  Cour, Sapp & Taskar (JMLR2011)
  - Multiple candidate classes are provided for each $\boldsymbol{x}_i$.
  - Complementary labels are the extreme case of partial labels given to all $c-1$ classes other than $\bar{y}_i$.
- Approach 2: Multi-label classification
  - Each $\boldsymbol{x}_i$ can belong to multiple classes.
  - Negative label for $\bar{y}_i$ and positives for the rest.
- We want a more direct approach!

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!

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

Theoretical Analysis

Ishida, Niu, Hu & Sugiyama (NIPS2017)

- Estimation error:

$$R(\boldsymbol{g}^*) - R(\widehat{\boldsymbol{g}}_{\mathrm{Comp}}) = \mathcal{O}_p\left(\frac{1}{\sqrt{n}}\right)$$

$$\boldsymbol{g}^* = \operatorname*{argmin}_{\boldsymbol{g}} R(\boldsymbol{g})$$

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

$$\widehat{\boldsymbol{g}}_{\mathrm{Comp}} = \operatorname*{argmin}_{\boldsymbol{g}} \widehat{R}_{\mathrm{Comp}}(\boldsymbol{g})$$

$$\widehat{R}_{\mathrm{Comp}}(\boldsymbol{g}) = \frac{1}{n}\sum_{i=1}^{n} \bar{L}\Big(\bar{y}_i, \boldsymbol{g}(\boldsymbol{x}_i)\Big)$$

- Optimal parametric convergence rate is attained!

Experiments

Correct classification rate: average (std)

5% t-test

Dataset	Class	Dim	# train	# test	Proposed	Partial-label	Multi-label
WAVEFORM1	$1 \sim 3$	21	1226	398	**85.8(0.5)**	**85.7(0.9)**	79.3(4.8)
WAVEFORM2	$1 \sim 3$	40	1227	408	**84.7(1.3)**	**84.6(0.8)**	74.9(5.2)
SATIMAGE	$1 \sim 7$	36	415	211	**68.7(5.4)**	60.7(3.7)	33.6(6.2)
PENDIGITS	$1 \sim 5$	16	719	336	**87.0(2.9)**	76.2(3.3)	44.7(9.6)
	$6 \sim 10$		719	335	**78.4(4.6)**	71.1(3.3)	38.4(9.6)
	even #		719	336	**90.8(2.4)**	76.8(1.6)	43.8(5.1)
	odd #		719	335	**76.0(5.4)**	67.4(2.6)	40.2(8.0)
	$1 \sim 10$		719	335	**38.0(4.3)**	33.2(3.8)	16.1(4.6)
DRIVE	$1 \sim 5$	48	3955	1326	**89.1(4.0)**	77.7(1.5)	31.1(3.5)
	$6 \sim 10$		3923	1313	**88.8(1.8)**	78.5(2.6)	30.4(7.2)
	even #		3925	1283	**81.8(3.4)**	63.9(1.8)	29.7(6.3)
	odd #		3939	1278	**85.4(4.2)**	74.9(3.2)	27.6(5.8)
	$1 \sim 10$		3925	1269	**40.8(4.3)**	32.0(4.1)	12.7(3.1)
LETTER	$1 \sim 5$	16	565	171	**79.7(5.3)**	**75.1(4.4)**	28.3(10.4)
	$6 \sim 10$		550	178	**76.2(6.2)**	66.8(2.5)	34.0(6.9)
	$11 \sim 15$		556	177	**78.3(4.1)**	67.4(3.3)	28.6(5.0)
	$16 \sim 20$		550	184	**77.2(3.2)**	68.4(2.1)	32.7(6.4)
	$21 \sim 25$		585	167	**80.4(4.2)**	75.1(1.9)	32.0(5.7)
	$1 \sim 25$		550	167	**5.1(2.1)**	**5.0(1.0)**	**5.2(1.1)**
USPS	$1 \sim 5$	256	652	166	**79.1(3.1)**	70.3(3.2)	44.4(8.9)
	$6 \sim 10$		542	147	**69.5(6.5)**	**66.1(2.4)**	37.3(8.8)
	even #		556	147	**67.4(5.4)**	**66.2(2.3)**	35.7(6.6)
	odd #		542	147	**77.5(4.5)**	69.3(3.1)	36.6(7.5)
	$1 \sim 10$		542	127	**30.7(4.4)**	26.0(3.5)	13.3(5.4)

Proposed method works well!

Incorporating Ordinary Labels

- Convert multiclass labeling into yes-no labeling:

http://www.softbank.jp/corp/group/sbr/news/press/2014/20141029_01/

https://www.bostondynamics.com/atlas

Is this Softbank Pepper? Yes! (ordinary label)	Is this iRobot Roomba? No! (complementary label)

- Use both of ordinary and complementary labels!

$$R(\boldsymbol{g}) = \alpha \mathbb{E}_{p(\boldsymbol{x},y)}\Big[L\Big(y, \boldsymbol{g}(\boldsymbol{x})\Big)\Big] + (1-\alpha)\mathbb{E}_{\bar{p}(\boldsymbol{x},\bar{y})}\Big[\bar{L}\Big(\bar{y}, \boldsymbol{g}(\boldsymbol{x})\Big)\Big]$$

$\alpha \in [0, 1]$

Experiments

$$R(f) = \alpha \mathbb{E}_{p(\boldsymbol{x},y)}\left[\mathcal{L}\Big(f(\boldsymbol{x}), y\Big)\right] + (1-\alpha)\left\{(c-1)\mathbb{E}_{\bar{p}(\boldsymbol{x},\bar{y})}\left[\bar{\mathcal{L}}\Big(f(\boldsymbol{x}), \bar{y}\Big)\right] + \text{Const.}\right\}$$

Dataset	Class	Dim	# train	# test	OL ($\alpha = 1$)	CL ($\alpha = 0$)	OL & CL ($\alpha = \frac{1}{2}$)
WAVEFORM1	$1 \sim 3$	21	413/826	408	85.3(0.8)	86.0(0.4)	**86.9(0.5)**
WAVEFORM2	$1 \sim 3$	40	411/821	411	82.7(1.3)	82.0(1.7)	**84.7(0.6)**
SATIMAGE	$1 \sim 7$	36	69/346	211	74.9(4.9)	70.1(5.6)	**81.2(1.1)**
PENDIGITS	$1 \sim 5$	16	144/575	336	**91.3(2.1)**	84.7(3.2)	**93.1(2.0)**
	$6 \sim 10$		144/575	335	**86.3(3.5)**	78.3(6.2)	**87.8(2.8)**
	even #		144/575	336	94.3(1.7)	91.0(4.3)	**95.8(0.6)**
	odd #		144/575	335	**85.6(2.0)**	75.9(3.1)	**86.9(1.1)**
	$1 \sim 10$		72/647	335	61.7(4.3)	41.1(5.7)	**66.9(2.0)**
DRIVE	$1 \sim 5$	48	780/3121	1305	92.1(2.6)	89.0(2.1)	**94.2(1.0)**
	$6 \sim 10$		795/3180	1290	**87.0(3.0)**	86.5(3.1)	**89.5(2.1)**
	even #		657/3284	1314	**91.4(2.9)**	81.8(4.6)	**91.8(3.3)**
	odd #		790/3161	1255	91.1(1.5)	86.7(2.9)	**93.4(0.5)**
	$1 \sim 10$		397/3570	1292	**75.2(2.8)**	40.5(7.2)	**77.6(2.2)**
LETTER	$1 \sim 5$	16	113/452	171	85.2(1.3)	77.2(6.1)	**89.5(1.6)**
	$6 \sim 10$		110/440	178	81.0(1.7)	77.6(3.7)	**84.6(1.0)**
	$11 \sim 15$		111/445	177	81.1(2.7)	76.0(3.2)	**87.3(1.6)**
	$16 \sim 20$		110/440	184	81.3(1.8)	77.9(3.1)	**84.7(2.0)**
	$21 \sim 25$		117/468	167	86.8(2.7)	81.2(3.4)	**91.1(1.0)**
	$1 \sim 25$		22/528	167	11.9(1.7)	6.5(1.7)	**31.0(1.7)**
USPS	$1 \sim 5$	256	130/522	166	83.8(1.7)	76.5(5.3)	**89.5(1.3)**
	$6 \sim 10$		108/434	147	79.2(2.1)	67.6(4.3)	**85.5(2.4)**
	even #		108/434	166	79.6(2.7)	67.4(4.4)	**84.8(1.4)**
	odd #		111/445	147	82.7(1.9)	72.9(6.2)	**87.3(2.2)**
	$1 \sim 10$		54/488	147	43.7(2.6)	28.5(3.6)	**59.3(2.2)**

5% t-test

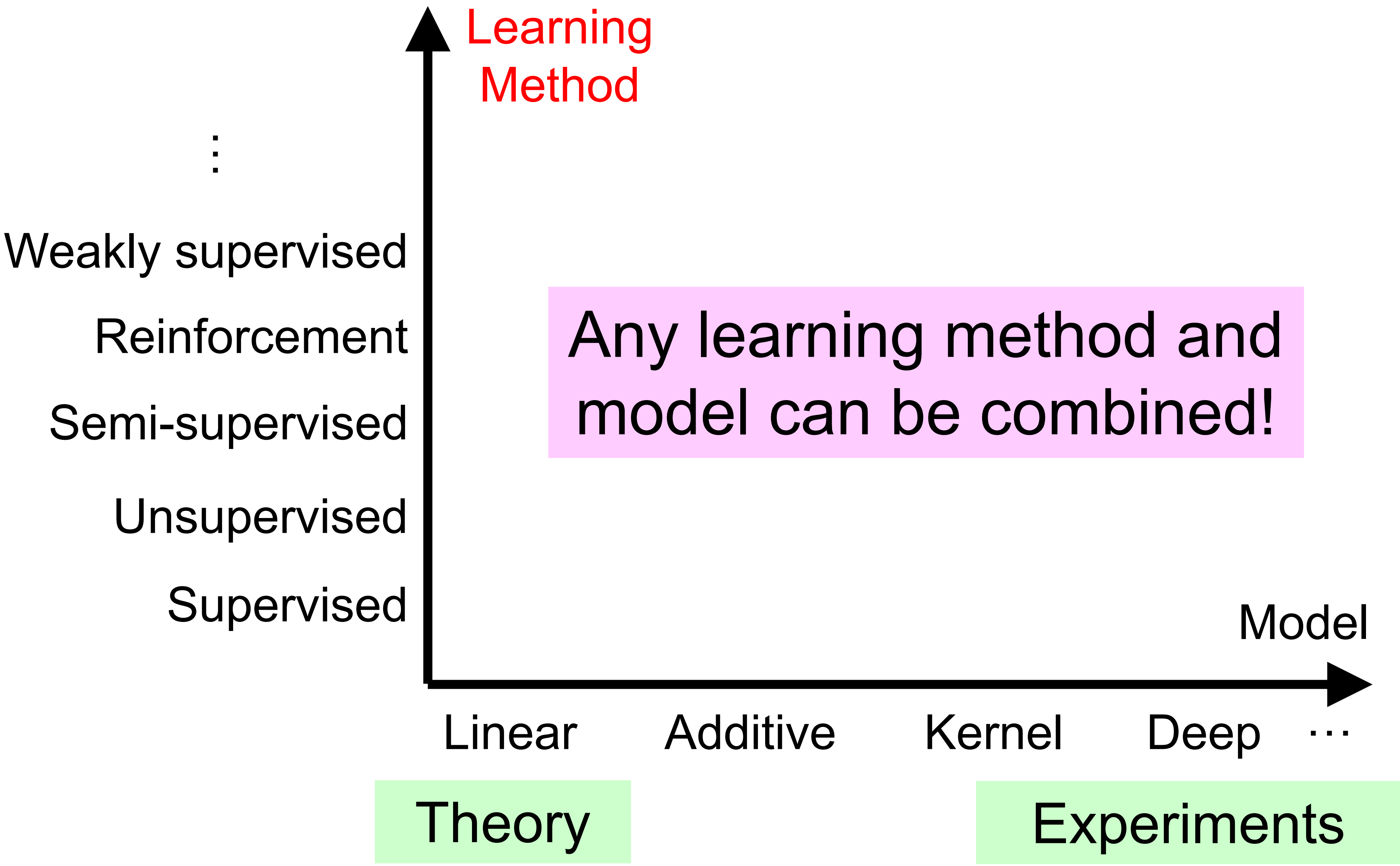
Incorporating complementary labels Improves the accuracy!

Contents

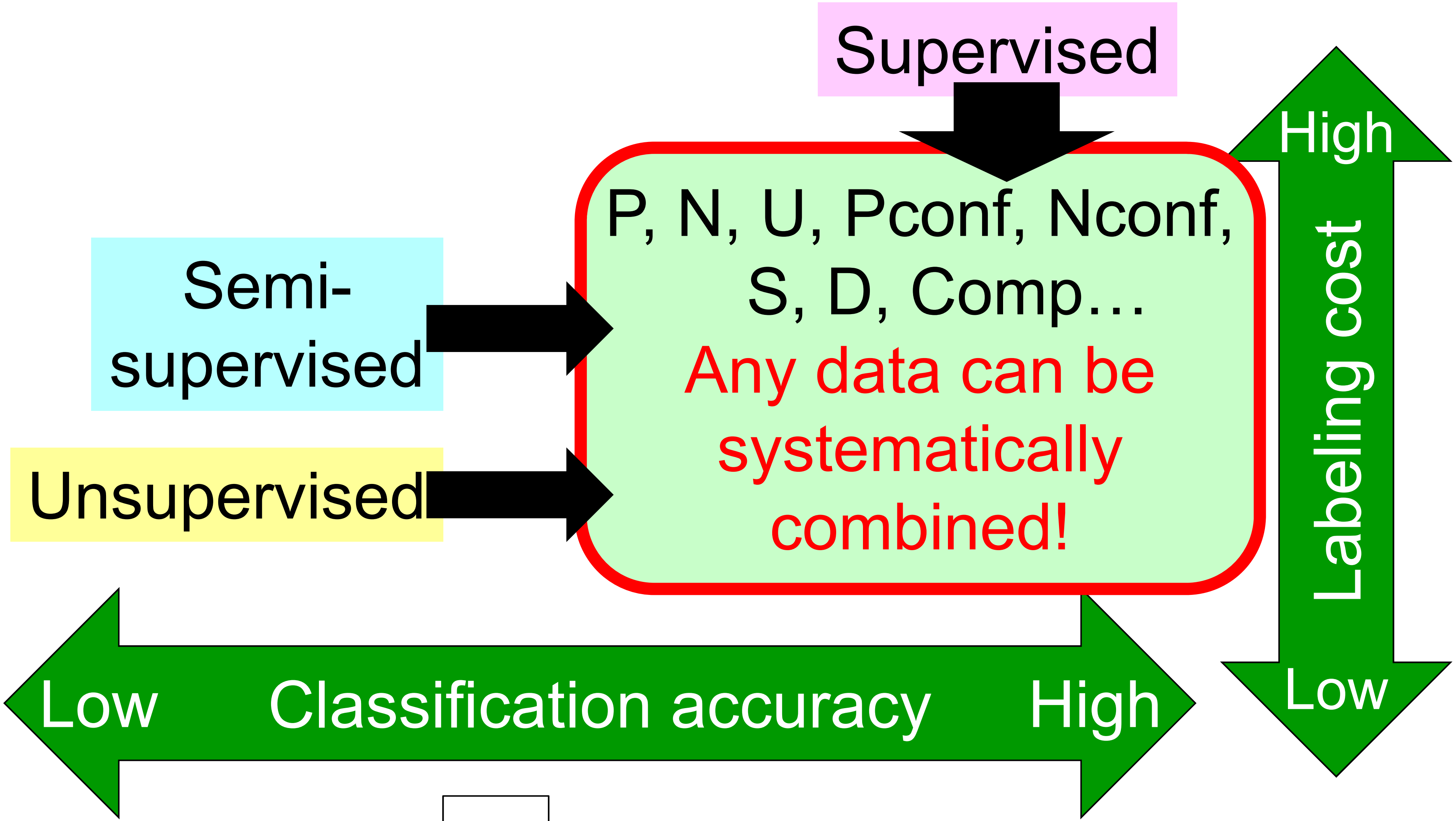
- P: Positive
- N: Negative
- U: Unlabeled
- Conf: Confidence
- S: Similar
- Comp: Complementary

Slides:
http://goo.gl/meiTwY

Model vs. Learning Methods



Learning from Weak Supervision



Sugiyama, Niu, Sakai & Ishida,
Machine Learning from Weak Supervision
MIT Press, 2020 (?)