MIT

Sep. 18, 2023

Machine Learning from Weak, Noisy, and Biased Supervision

Masashi Sugiyama

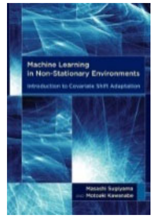
RIKEN Center for Advanced Intelligence Project/
The University of Tokyo, Japan

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

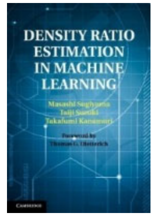
About Myself

- **Masashi Sugiyama**:
  - Director: RIKEN AIP, Japan
  - Professor: University of Tokyo, Japan
  - Consultant: several local startups
- **Interests: Machine learning (ML)**
  - ML theory & algorithm →
  - ML applications (signal, image, language, brain, robot, mobility, advertisement, biology, medicine, education…)
- **Academic activities:**
  - Program Chairs for NeurIPS2015, AISTATS2019, ACML2010/2020…
  - Keynote speaker at ICLR2023.

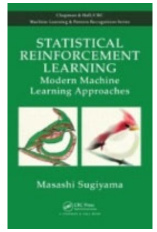
Sugiyama & Kawanabe, Machine Learning in Non-Stationary Environments, MIT Press, 2012



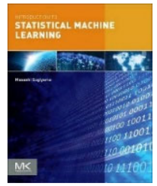
Sugiyama, Suzuki & Kanamori, Density Ratio Estimation in Machine Learning, Cambridge University Press, 2012



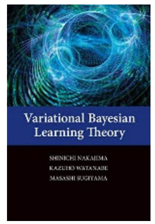
Sugiyama, Statistical Reinforcement Learning, Chapman and Hall/CRC, 2015



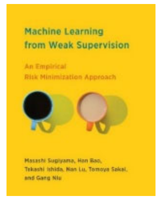
Sugiyama, Introduction to Statistical Machine Learning, Morgan Kaufmann, 2015



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



Sugiyama, Bao, Ishida, Lu, Sakai & Niu. Machine Learning from Weak Supervision, MIT Press, 2022.



What is “RIKEN”?

- Name in Japanese: 理化学研究所
  - Pronounced as: rikagaku kenkyusho
  - Meaning: Physics and Chemistry Research Institute
- Acronym in Japanese: 理研 (RIKEN)

Brief History


Shibusawa Eiichi

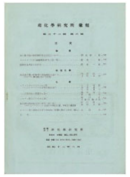

Okochi Masatoshi


Suzuki Umetaro


Nishina Yoshio


Yukawa Hideki

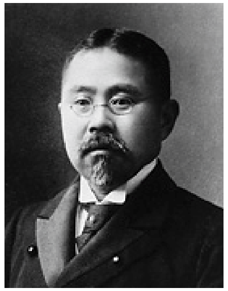

Tomonaga Shinichiro




Noyori Ryoji


Matsumoto Hiroshi


Gonokami Makoto


Kikuchi Dairoku


Takamine Jokichi

1917– RIKEN private foundation

RIKEN Konzern
In 1939, 63 companies and 121 plants

1948– Scientific Research Institute Ltd. (KAKEN)

1958– RIKEN public corporation

1967 moved to Wako

2003– RIKEN Independent Administrative Institution

2015– RIKEN National Research and Development Institute

2017 The centennial celebration

1984 Tsukuba
1990 Sendai
1993 Nagoya
1997 Harima
1998 Yokohama
2000 Kobe
2011 Osaka
2016 Keihanna

2017
Wako: 50^{th} Anniversary
Harima: 20^{th} Anniversary

Office and Research

2900 researchers
500 admin staffs

RIKEN Information R&D and Strategy Headquarters

Informatics

Spring-8 Center

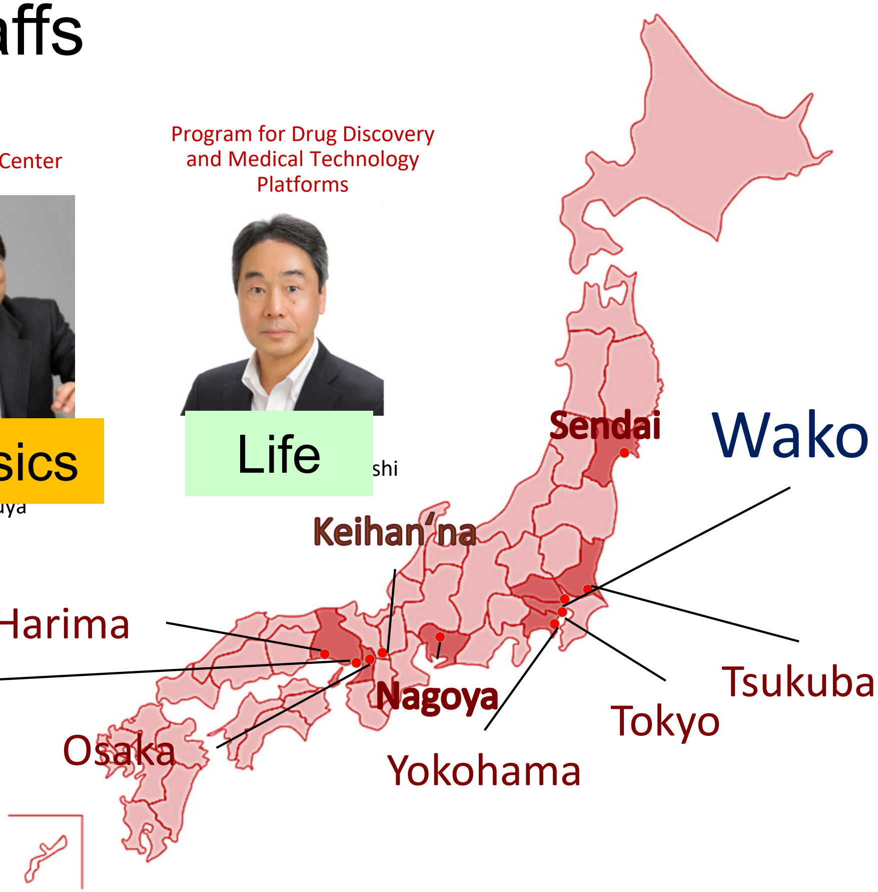
Physics

Tetsuya

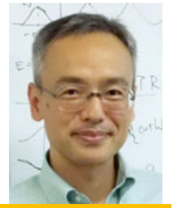
Program for Drug Discovery and Medical Technology Platforms

Life

shi



Center for Quantum Computing



Physics

Yasunobu

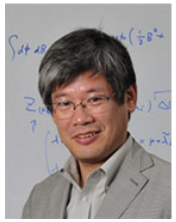
Center for Emergent Matter Science

Physics

Nishina Center for Accelerator-Based Science

Physics

Interdisciplinary Theoretical and Mathematical Sciences Program



Physics

Center for Advanced Photonics

Physics

Center for Brain Science

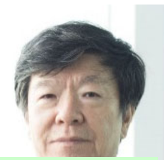
Life

Center for Computational Science

Informatics

Director
Dr.Matsuoka Satoshi

Center for Biosystems Dynamics Research



Life

Director
Dr. Nishida Eisuke

Center for Integrative Medical Sciences

Life

Director
Dr. Yamamoto Kazuhiko

Center for Sustainable Resource Science

Life

Director
Dr. Saito Kazuki

Center for Advanced Intelligence Project

Informatics

Director
Dr. Sugiyama Masashi

BioResource Research Center

Life

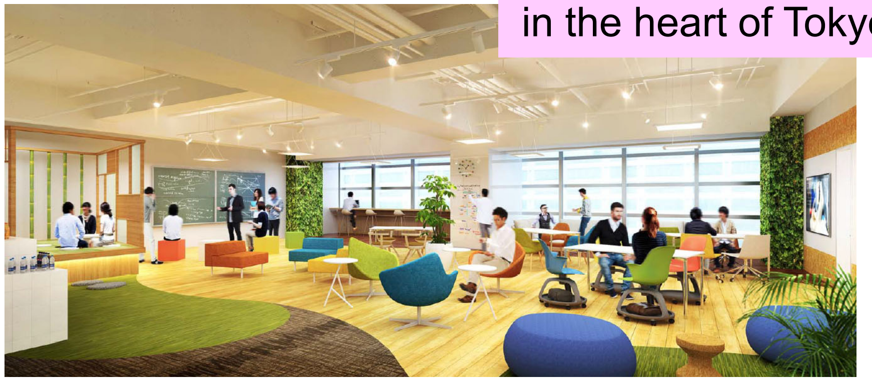
Director
Dr. Shiroishi Toshihiko

What is RIKEN-AIP?

- RIKEN founded Center for Advanced Intelligence Project (AIP) in 2016, under Ministry of Education, Culture, Sports, Science and Technology (MEXT):
  - 130 employed researchers (40% international, 25% female)
  - 250 visiting researchers
  - 130 domestic students
  - 140 international interns (total)
  - 40+ international collaboration partners
  - 40+ industry projects

MEXT
MINISTRY OF EDUCATION, CULTURE, SPORTS, SCIENCE AND TECHNOLOGY-JAPAN

Main office
in the heart of Tokyo

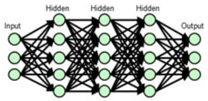


Distributed offices
across Japan

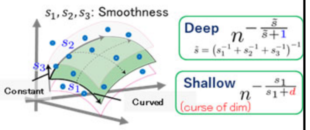
Selected Research of RIKEN-AIP

Developing New AI Technology

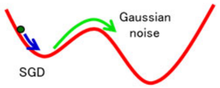
- **Theory of deep learning**:
  - Better prediction than shallow learning
  - No curse of dimensionality
  - Global optimization

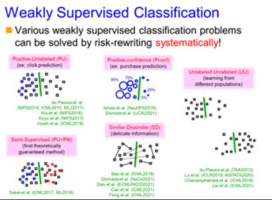


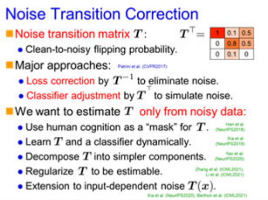
$$\mathbb{E}[\|\hat{f}_T - f^*\|_{L_2}^2] \le \epsilon_M + O(T^{-\frac{2r\beta}{2r\beta+1}})$$

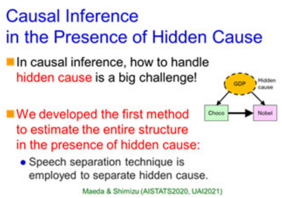


- **Developing new methods**:
  - Weakly supervised learning
  - Noise robust learning
  - Causal inference



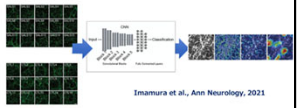


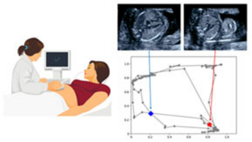




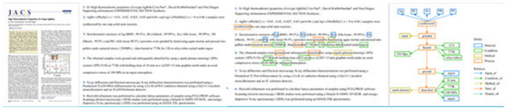
Accelerating Scientific Research

- **Medical science**:
  - Prostate/pancreatic cancer detection
  - ALS early diagnosis
  - Fetal heart screening
  - Colonoscopy

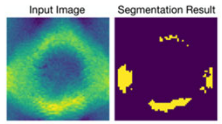


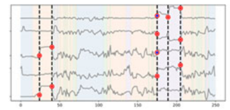


- **Material science**:
  - Database creation with text mining



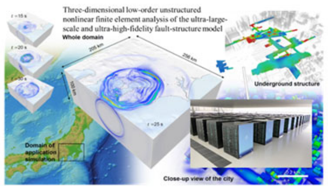
- **Data-driven science**:
  - Selective inference for reliability evaluation

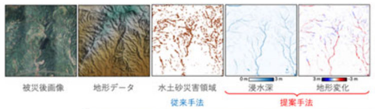




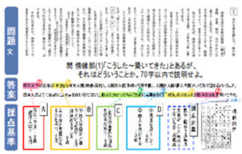
Solving Socially Critical Problems

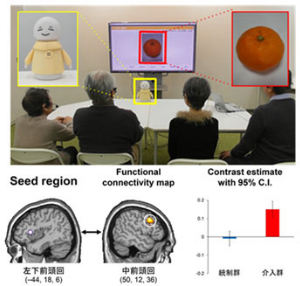
- **Natural disaster**:
  - Fugaku-based earthquake simulation
  - Remote sensing disaster analysis





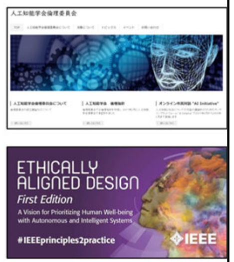
- **Elderly healthcare**:
  - Chat-robot-guided cognitive function improvement
- **Education**:
  - Automatic essay evaluation
  - Interactive essay writing support

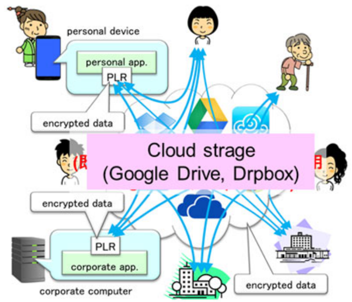


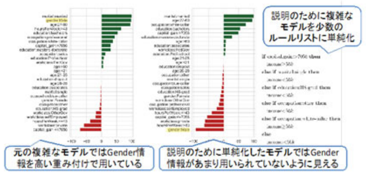


Studying AI-ELSI

- **AI Ethical guidelines**:
  - Japanese Society for AI, Ministry of Internal Affairs and Communications, Cabinet Office
  - IEEE, G20, OECD
- **Personal data management**:
  - Individual-based accessibility control system
- **AI security and reliability**:
  - Adversarial attack/defense
  - Fairness faking/guarantee

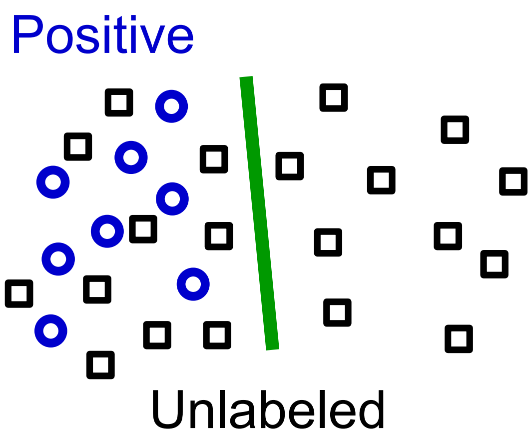


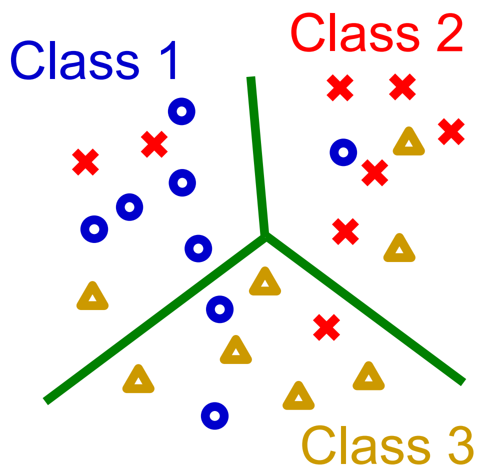


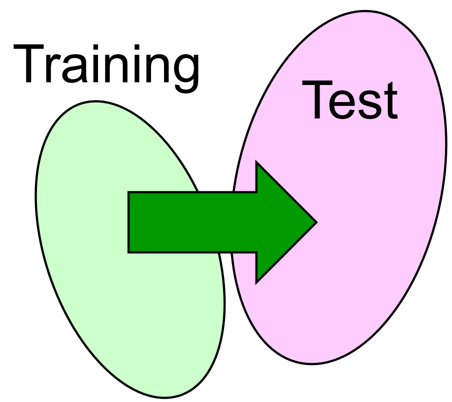


Reliable Machine Learning

- Reliability of machine learning systems can be degraded by various factors:
  - Insufficient information: weak supervision.
  - Label noise: human error, sensor error.
  - Data bias: changing environments, privacy.
- Improving the reliability is an urgent challenge!



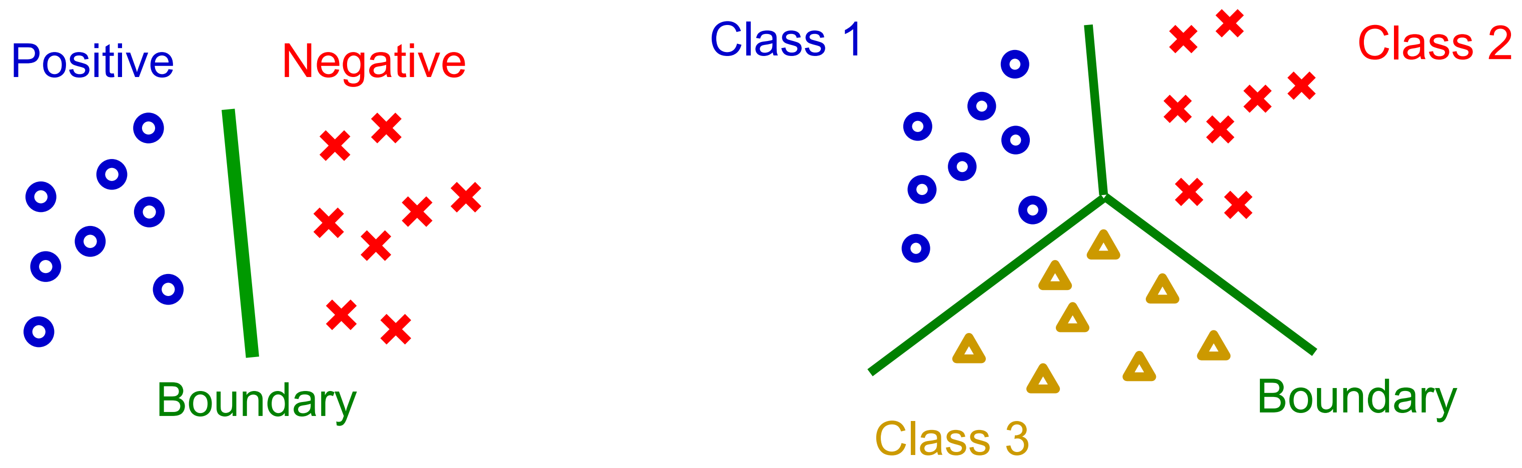




Contents

Weakly Supervised Classification

- Supervised classification from big labeled data is successful: speech, image, language, …



- However, there are many applications where big labeled data is not available:
  - Medicine, disaster, robot, brain, …

- We want to utilize "weak" supervision that can be collected easily!

Positive-Unlabeled (PU) Classification

Li+ (IJCAI2003)

- Given: PU samples (no N samples).

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

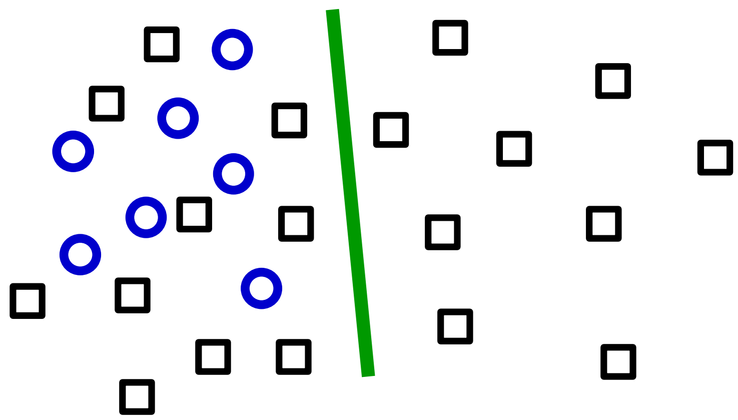
- Goal: Obtain a classifier minimizing the PN risk.

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

$\mathbb{E}$: expectation ℓ : loss $y = \{+1, -1\}$

Example: Ad click prediction

- Clicked ad: User likes it → P
- Unclicked ad: User dislikes it or User likes it but doesn't have time to click it → U (=P or N)



PU Unbiased Risk Estimation

du Plessis+ (NeurIPS2014, ICML2015)

- Decompose the risk:

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

$\pi = p(y=+1)$: Class prior (assumed known) →

Scott+ (AISTATS2009)
Ramaswamy+ (ICML2016)
du Plessis+ (MLJ2017)
Yao+ (ICLR2022)

- Without N data, $R^-(f)$ can not be estimated directly:
  - Eliminate the expectation over N data as

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

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

- Unbiased risk estimator:

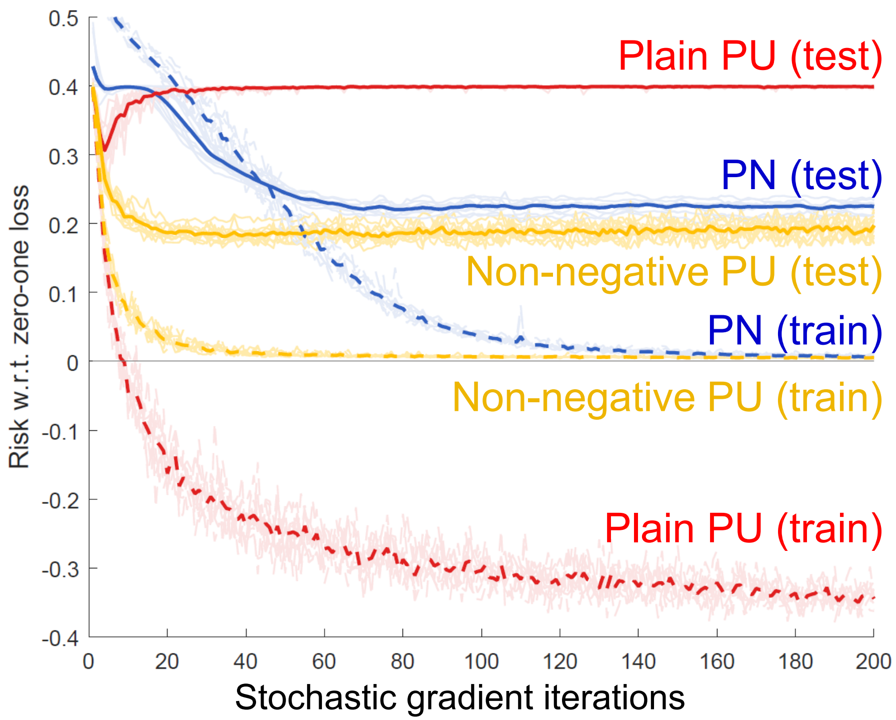
$$\widehat{R}_{\mathrm{PU}}(f) = \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(+1, f(\boldsymbol{x}_i^{\mathrm{P}})\Big) + \frac{1}{n_{\mathrm{U}}}\sum_{j=1}^{n_{\mathrm{U}}}\ell\Big(-1, f(\boldsymbol{x}_j^{\mathrm{U}})\Big) - \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}}\ell\Big(-1,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

Non-Negative Risk Correction

Kiryo+ (NeurIPS2017) , Lu+ (AISTATS2020)

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

- Risk for N data: $R^-(f) = \mathbb{E}_{p(\boldsymbol{x})}\Big[\ell\Big(-1, f(\boldsymbol{x})\Big)\Big] - \pi \mathbb{E}_{p(\boldsymbol{x}|y=+1)}\Big[\ell\Big(-1, f(\boldsymbol{x})\Big)\Big]$
- Empirical estimate: $\widehat{R}^-_{\mathrm{PU}}(f) = \frac{1}{n_{\mathrm{U}}}\sum_{i=1}^{n_{\mathrm{U}}} \ell\Big(-1, f(\boldsymbol{x}_i^{\mathrm{U}})\Big) - \frac{\pi}{n_{\mathrm{P}}}\sum_{i=1}^{n_{\mathrm{P}}} \ell\Big(-1, f(\boldsymbol{x}_i^{\mathrm{P}})\Big)$
- When loss is non-negative:
  - True $R^-(f)$ is non-negative.
  - But empirical estimate can be negative!

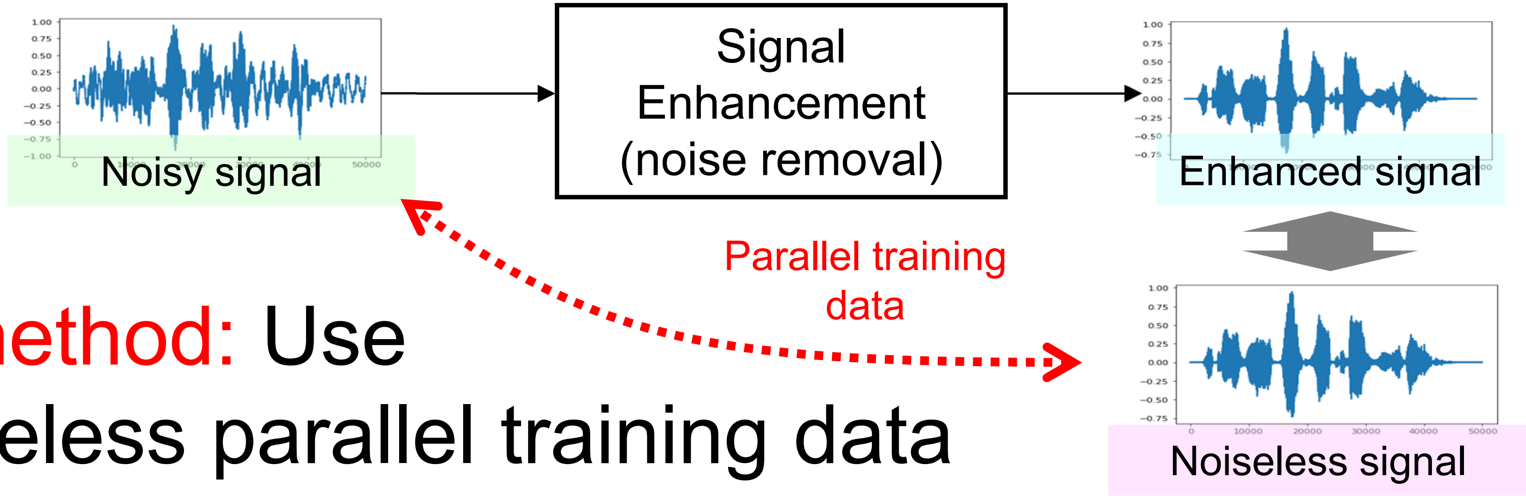


- Non-negative correction:

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

Signal Enhancement by PU Classification

Ito & Sugiyama (ICASSP2023, Best Paper Award)

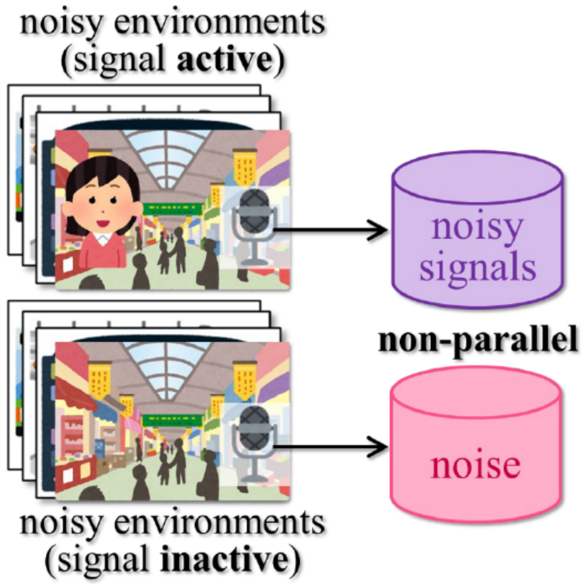


- **Existing method:** Use noisy/noiseless parallel training data
  - In practice, use synthetic data
    → Do not generalize well in reality.

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

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

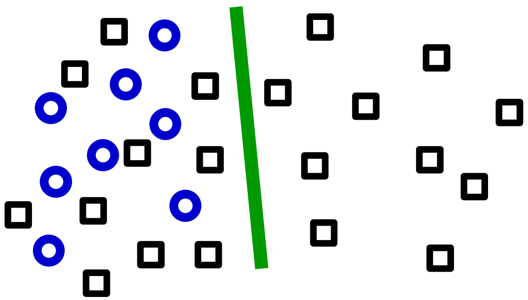
Non-parallel: Proposed, MixIT
Parallel: Supervised



Various Extensions (Binary)

- Similar unbiased risk estimation is possible!

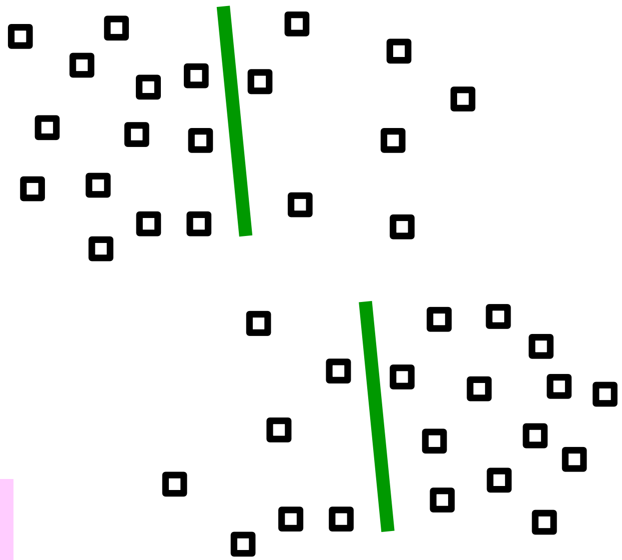
Positive-Unlabeled (PU)



du Plessis+ (NeurIPS2014, ICML2015, MLJ2017), Niu+ (NeurIPS2016), Kiryo+ (NeurIPS2017), Hsieh+ (ICML2019)

Click prediction

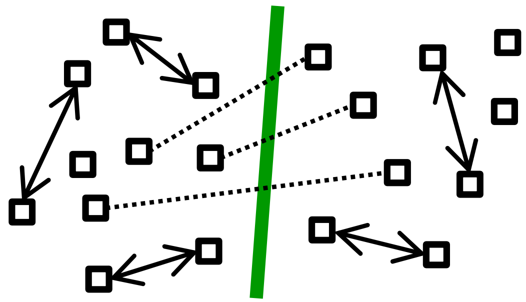
Unlabeled-Unlabeled (UU)



du Plessis+ (TAAI2013), Lu+ (ICLR2019, AISTATS2020), Charoenphakdee+ (ICML2019), Lei+ (ICML2021)

Different populations

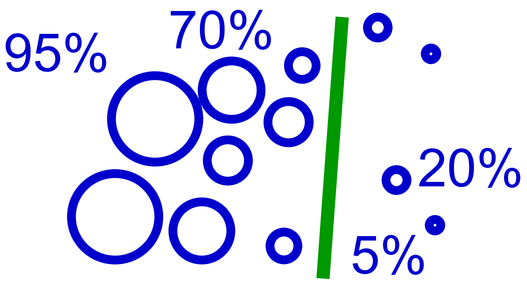
Similar-Dissimilar (SD)



Bao+ (ICML2018), Shimada+ (NeCo2021), Dan+ (ECMLPKDD2021), Cao+ (ICML2021), Feng+ (ICML2021)

Sensitive prediction

Positive-confidence (Pconf)

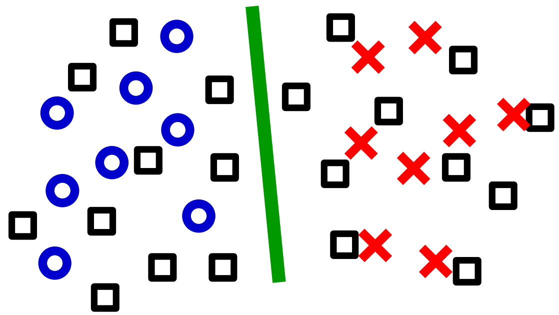


Ishida+ (NeurIPS2018), Shinoda+ (IJCAI2021)

Purchase prediction

Semi-supervised classification without manifold/clusters →

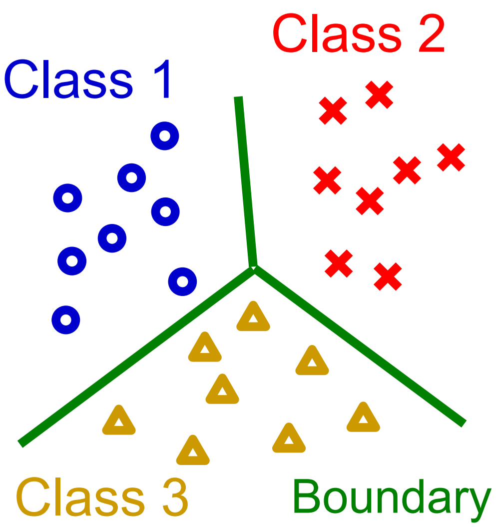
Positive-Negative-Unlabeled (PNU)



Sakai+ (ICML2017, ML2018)

Various Extensions (Multiclass)

- Labeling patterns in multi-class problems is even more painful.

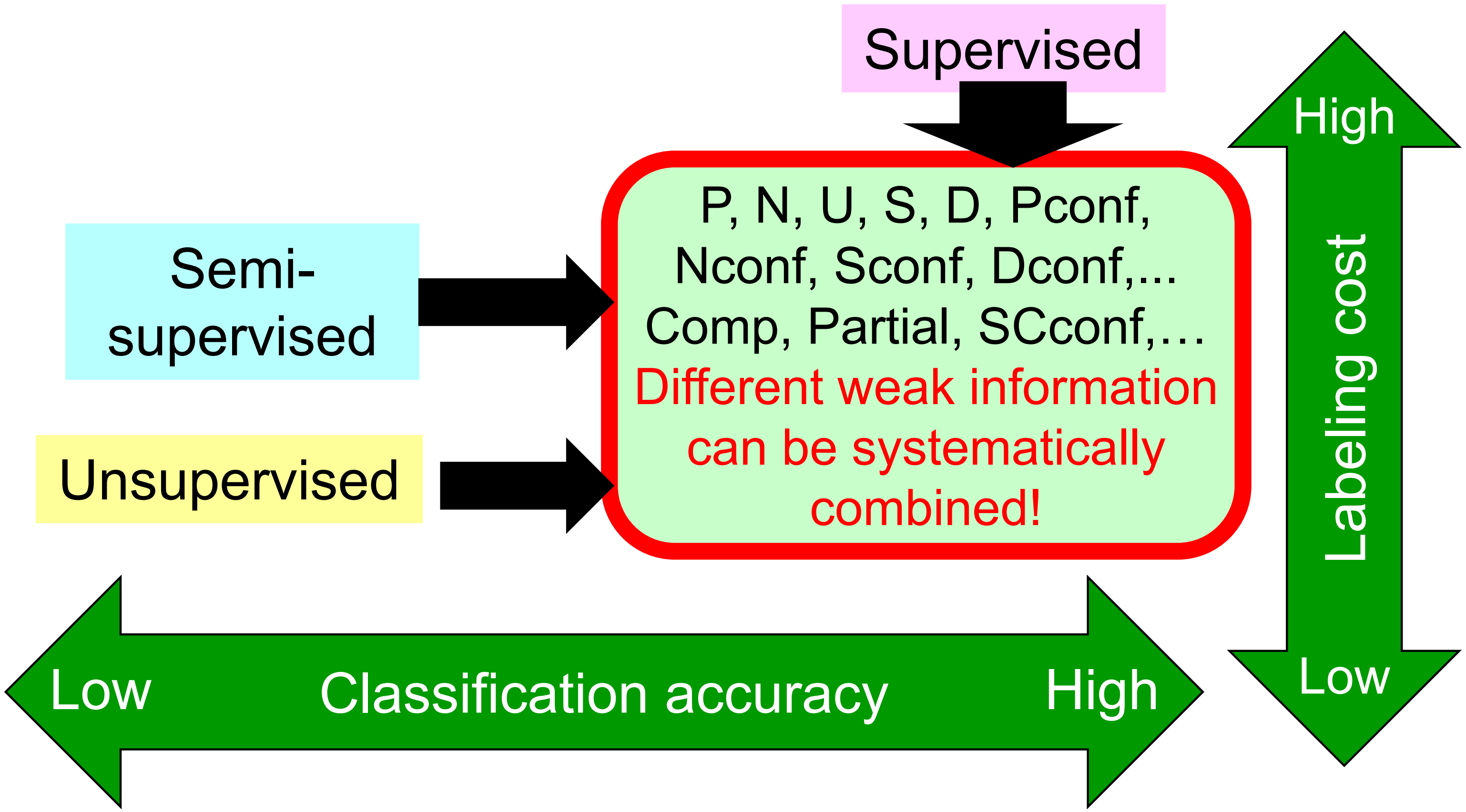


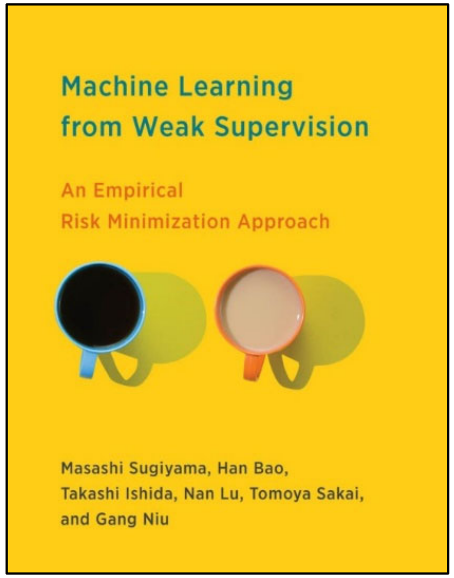
- Multi-class weak-labels:
  - Complementary label: Specifies a class that a pattern does not belong to ("not 1"). Ishida+ (NeurIPS2017, ICML2019), Chou+ (ICML2020)
  - Partial label: Specifies a subset of classes that contains the correct one ("1 or 2"). Feng+ (ICML2020, NeurIPS2020), Lv+ (ICML2020)
  - Single-class confidence: Cao+ (arXiv2021) One-class data with full confidence ("1 with 60%, 2 with 30%, and 3 with 10%")

- Similar unbiased risk estimation is possible!

Summary: Weakly Supervised Learning

- Empirical risk minimization framework for weakly supervised learning:
  - Any loss, classifier, and optimizer can be used.



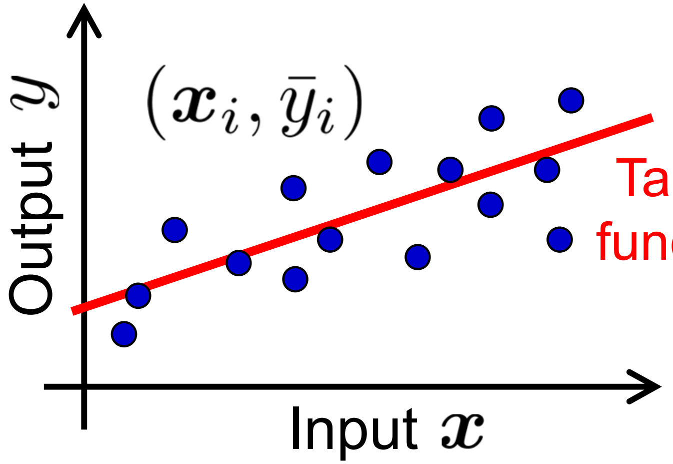


Sugiyama, Bao, Ishida, Lu, Sakai & Niu, MIT Press, 2022.

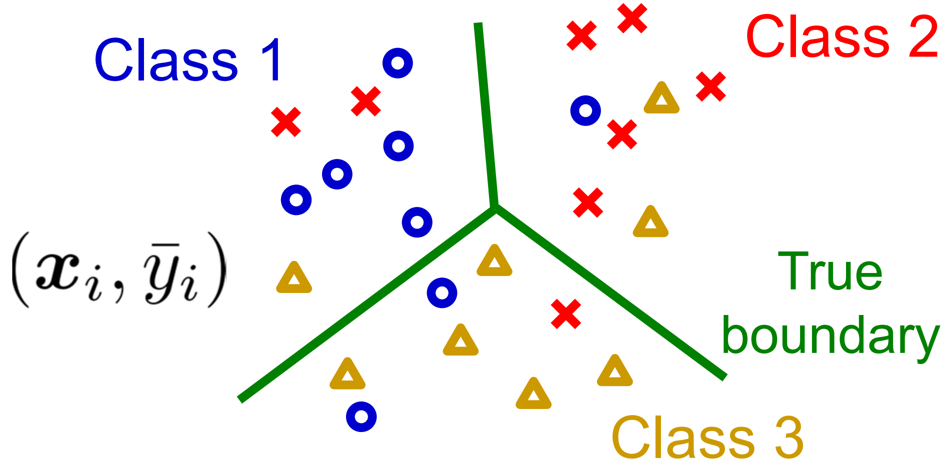
Contents

Supervised Learning with Noisy Output

Regression (additive noise)



Classification (label flipping noise)



$$\min_{\boldsymbol{g}} \sum_{i=1}^{n} \ell\Big(\bar{y}_i, \boldsymbol{g}(\boldsymbol{x}_i)\Big)$$

ℓ: loss

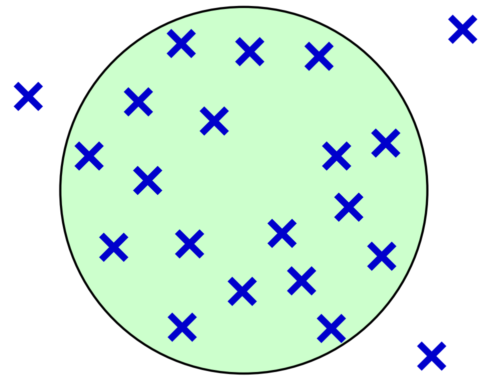
$\bar{y}$: noisy output

$\boldsymbol{g}$: probabilistic classifier

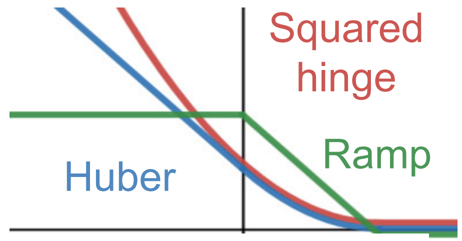
- Hasn't such a classic problem been solved?
  - Regression: Yes, noisy big data yield consistency.
  - Classification: Specific noise reduction mechanism is needed to achieve consistency!

Classical Approaches

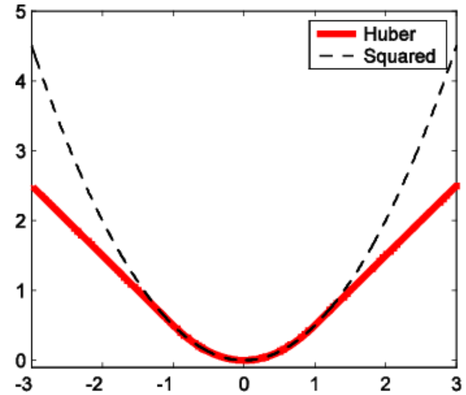
- **Unsupervised outlier removal:**
  - Substantially more difficult than classification.



- **Robust loss:**
  - Works well for regression, but limited effectiveness for classification.

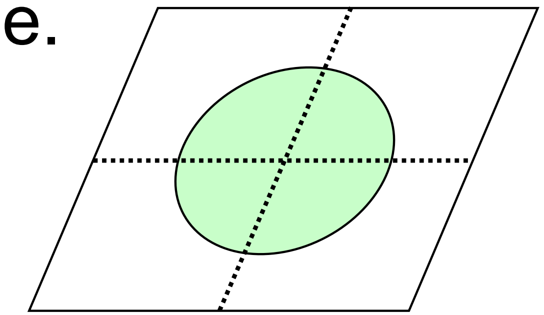


Classification margin

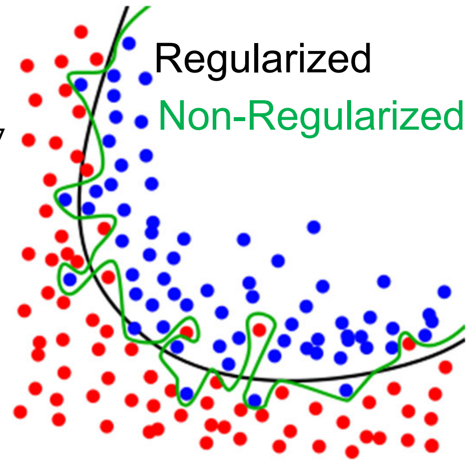


Residual

- **Regularization:**
  - Effective in suppressing overfitting, but too smooth for strong noise.



ℓ_2-regularization



https://en.wikipedia.org/wiki/Overfitting

- **Need new approaches!**

Correction with Noise Transition

- **Noise transition matrix $\boldsymbol{T}$:**
  - Clean-to-noisy flipping probability.

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0.1 & 0.8 & 0.1 \\ 0.5 & 0.5 & 0 \end{bmatrix}$$

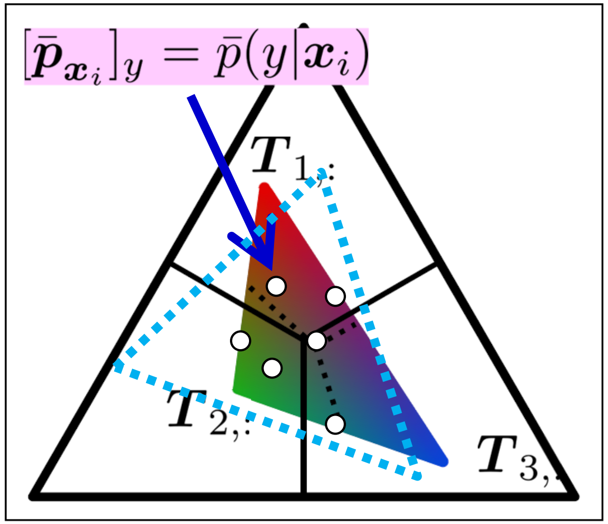
- **Major approaches:** Patrini+ (CVPR2017)
  - Classifier adjustment by $\boldsymbol{T}^{\top}$ to simulate noise.
  - Loss correction by $\boldsymbol{T}^{-1}$ to eliminate noise.

- **We want to estimate $\boldsymbol{T}$ only from noisy data:**
  - Use human cognition as a “mask” for $\boldsymbol{T}$. Han+ (NeurIPS2018)
  - Reduce estimation error of $\boldsymbol{T}$. Xia+ (NeurIPS2019), Yao+ (NeurIPS2020)
  - Learn $\boldsymbol{T}$ and classifier simultaneously. Zhang+ (ICML2021)
  - Estimate $\boldsymbol{T}$ under weaker conditions. Li+ (ICML2021)

Volume Minimization

Li+ (ICML2021)

- Noisy training data $\{(\boldsymbol{x}_i, \bar{y}_i)\}_{i=1}^n$ can be mapped in the simplex formed by noise transition matrix $\boldsymbol{T}$.

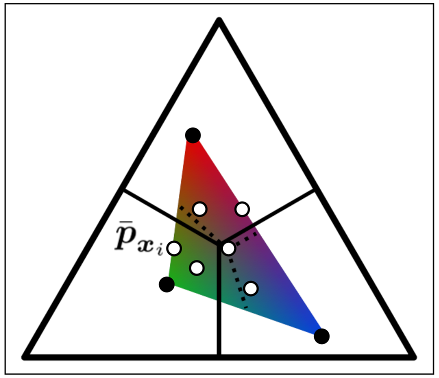


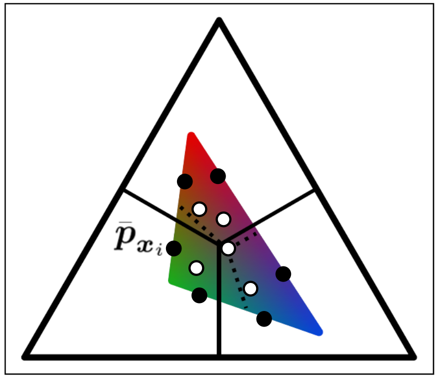
- Minimizing the volume of the simplex can give a solution:

$$\min_{\boldsymbol{T}', \boldsymbol{g}} \sum_{i=1}^{n} \ell(\bar{y}_i, \boldsymbol{T}'^{\top} \boldsymbol{g}(\boldsymbol{x}_i)) + \lambda \log \det(\boldsymbol{T}')$$

$\lambda > 0$

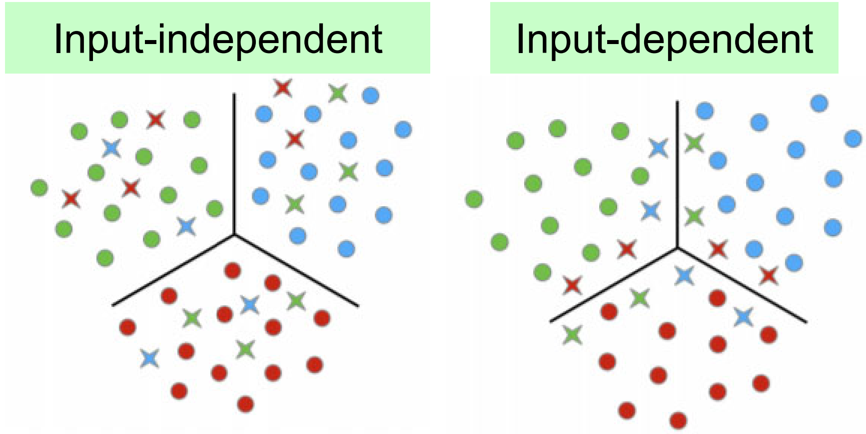
  - With noiseless labels, we can find the true $\boldsymbol{T}$.
  - Even without noiseless labels, "sufficiently scattered" training data allow identification of the true $\boldsymbol{T}$!





Beyond Input-Independent Noise

- Real-world noise may be input-dependent:
  - E.g., noise level is high near the boundary.



- Modeling input-dependent noise: $T_{y,\bar{y}}(\boldsymbol{x}) = \bar{p}(\bar{y}|y, \boldsymbol{x})$
  - Extremely challenging to estimate the noise transition matrix function!

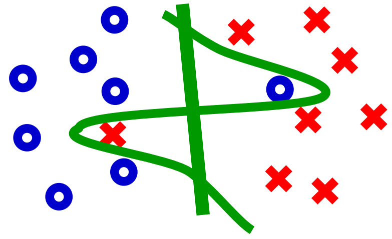
- Exploring heuristic solutions:
  - Parts-based estimation. Xia+ (NeurIPS2020)
  - Use of additional confidence scores. Berthon+ (ICML2021)
  - Manifold regularization. Cheng+ (CVPR2022)

Co-teaching

Memorization of neural nets:

Arpit+ (ICML2017)
Zhang+ (ICLR2017)

- Stochastic gradient descent fits clean data faster.
- However, naïve early stopping does not work well.



"Co-teaching" between two neural nets:

- Teach small-loss data each other.

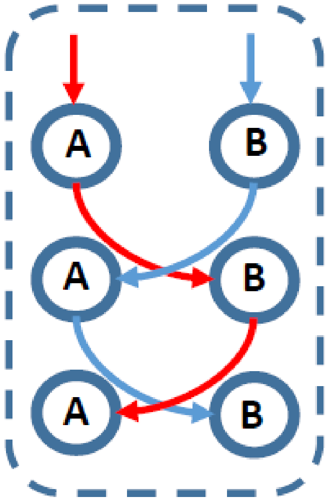
Han+ (NeurIPS2018)

- Teach only disagreed data.

Yu+ (ICML2019)

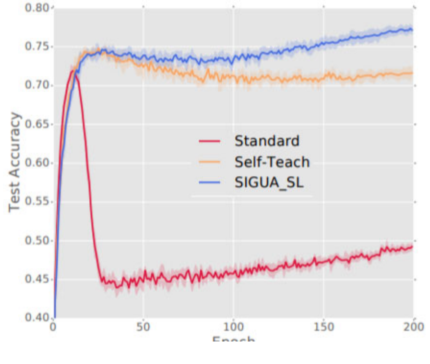
- Gradient ascent for large-loss data.

Han+ (ICML2020)



No theory but very robust in experiments:

- Works well even if 50% random label flipping!



Summary: Noisy-Label Learning

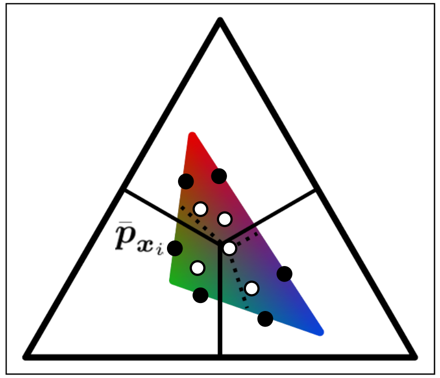
- Explicit treatment of label noise is necessary:
  - Loss correction by noise transition is promising.

$$T_{y,\bar{y}} = \bar{p}(\bar{y}|y)$$

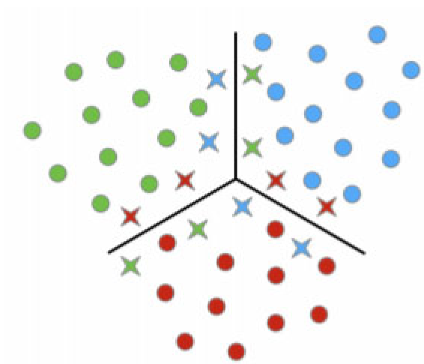
- However, noise transition is generally non-identifiable:

$$\boldsymbol{T}^\top \boldsymbol{p} = \boldsymbol{T}_2^\top (\boldsymbol{T}_1^\top \boldsymbol{p}) \qquad \boldsymbol{T} = \boldsymbol{T}_1 \boldsymbol{T}_2$$

  - Recent development allows consistent estimation under mild assumptions.



- Real-world noise is often input-dependent:
  - Heuristic solutions have been developed.
  - Further theoretical development is needed.

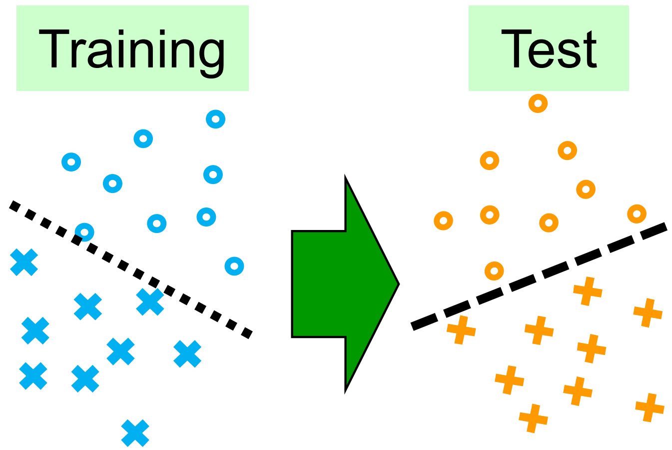


Contents

1. Weakly Supervised Learning
2. Noisy-Label Learning
3. Transfer Learning
4. Towards More Reliable Learning

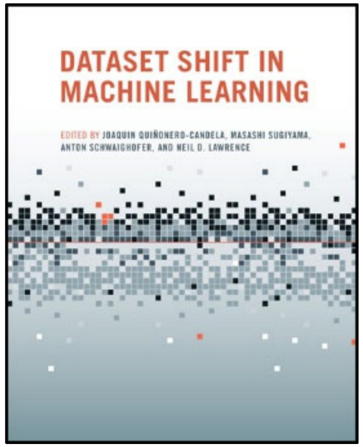
Transfer Learning

- Training and test data often follow different distributions, due to
  - changing environments,
  - sample selection bias (privacy).



- Transfer learning:
  - Train a test-domain predictor using training data from different domains.

NIPS Workshop on Learning when Test and Training Inputs Have Different Distributions, Whistler 2006



Quiñonero-Candela, Sugiyama, Schwaighofer & Lawrence (MIT Press 2009)

Basics: Importance-Weighted Training

- Covariate shift: Only input distributions change.

$$p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x})$$
$$p_{\mathrm{tr}}(y|\boldsymbol{x}) = p_{\mathrm{te}}(y|\boldsymbol{x})$$

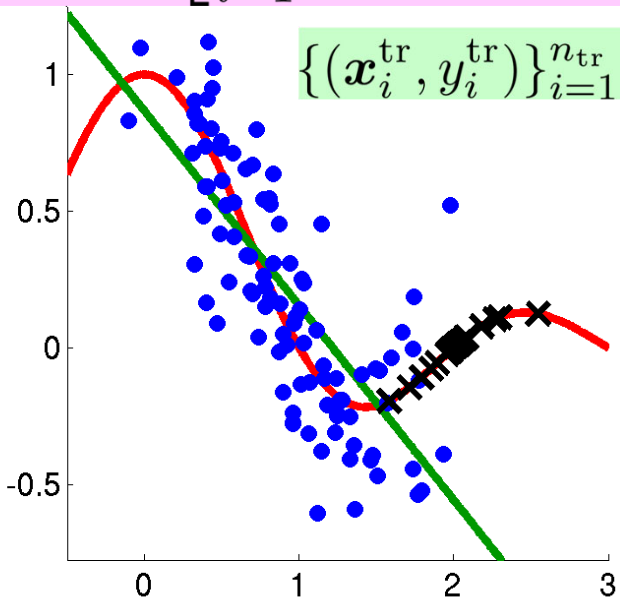
$\boldsymbol{x}$: Input

y: Output

Shimodaira (JSPI2000)

$$\underset{f}{\operatorname{argmin}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right]$$

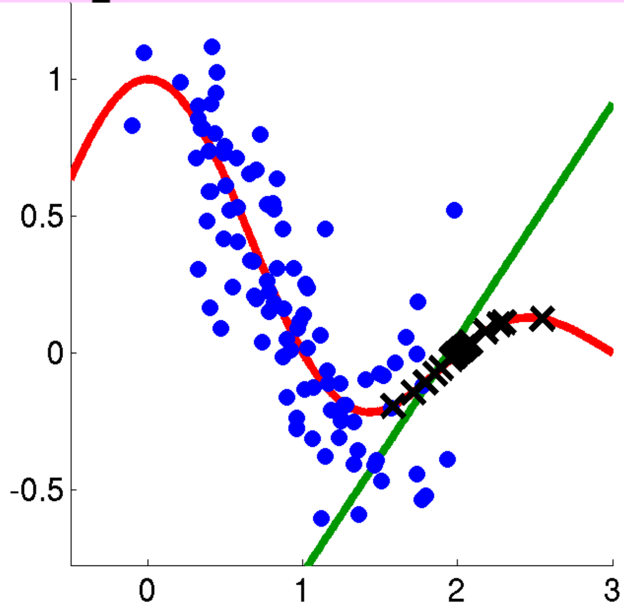
$$\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y)$$



Ordinary training is not consistent

Importance

$$\underset{f}{\operatorname{argmin}} \left[\sum_{i=1}^{n_{\mathrm{tr}}} \frac{p_{\mathrm{te}}(\boldsymbol{x}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\boldsymbol{x}_i^{\mathrm{tr}})} \ell(f(\boldsymbol{x}_i^{\mathrm{tr}}), y_i^{\mathrm{tr}}) \right]$$



Importance-weighted training is consistent

Direct Importance Estimation

- **Given**: training and test input data

$$\{\boldsymbol{x}_i^{\mathrm{tr}}\}_{i=1}^{n_{\mathrm{tr}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}) \qquad \{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{te}}(\boldsymbol{x})$$

- **Kernel mean matching:** Huang+ (NeurIPS2006)
  - Match the means of $r(\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})$ and $p_{\mathrm{te}}(\boldsymbol{x})$ in RKHS $\mathcal{H}$.

$$\min_{r\in\mathcal{H}} \left\| \int K(\boldsymbol{x},\cdot)p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \int K(\boldsymbol{x},\cdot)r(\boldsymbol{x})p_{\mathrm{tr}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \right\|_{\mathcal{H}}^2$$

$K(\boldsymbol{x},\cdot)$: kernel

- **Least-squares importance fitting (LSIF)**: Kanamori+ (NeurIPS2008)
  - Fit a model $r(\boldsymbol{x})$ to $\frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}$ by least squares:

$$\operatorname*{argmin}_r \left[\int \left(r(\boldsymbol{x}) - \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})} \right)^2 p_{\mathrm{tr}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \right]$$

$$= \operatorname*{argmin}_r \left[\int r(\boldsymbol{x})^2 p_{\mathrm{tr}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - 2\int r(\boldsymbol{x})p_{\mathrm{te}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \right]$$

- They do not estimate $p_{\mathrm{tr}}(\boldsymbol{x}), p_{\mathrm{te}}(\boldsymbol{x})$, but $\frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}$ directly!

Classical Two-Step Adaptation

1. Importance weight estimation
   (e.g., least-squares importance fitting):

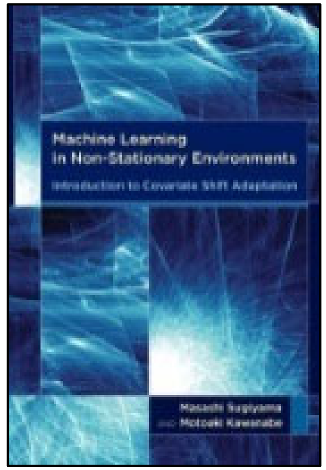
   Kanamori+ (JMLR2009)

$$\widehat{w} = \mathop{\mathrm{argmin}}_{w} \widehat{\mathbb{E}}_{p_{\mathrm{tr}}(\boldsymbol{x})} \left[\left(w(\boldsymbol{x}) - \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})} \right)^2 \right]$$

2. Weighted predictor training:

$$\widehat{f} = \mathop{\mathrm{argmin}}_{f} \widehat{\mathbb{E}}_{p_{\mathrm{tr}}(\boldsymbol{x},y)} [\widehat{w}(\boldsymbol{x}) \ell(f(\boldsymbol{x}), y)]$$

- However, estimation error in Step 1 is not taken into account in Step 2.
  - We want to integrate these two steps!



Sugiyama & Kawanabe (MIT Press 2012)

Joint Weight-Predictor Optimization

Zhang+ (ACML2020, SNCS2021)

- Given: Labeled training data and unlabeled test data

$$\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y) \qquad \{\boldsymbol{x}_j^{\mathrm{te}}\}_{j=1}^{n_{\mathrm{te}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{te}}(\boldsymbol{x})$$

- Joint minimization of a risk upper bound:

$$\min_{w \geq 0, f \in \mathcal{F}} J_{\ell'}(w, f) \qquad \frac{1}{2} R_\ell(f)^2 \leq J_{\ell'}(w, f) \qquad \ell \leq 1, \ell' \geq \ell$$

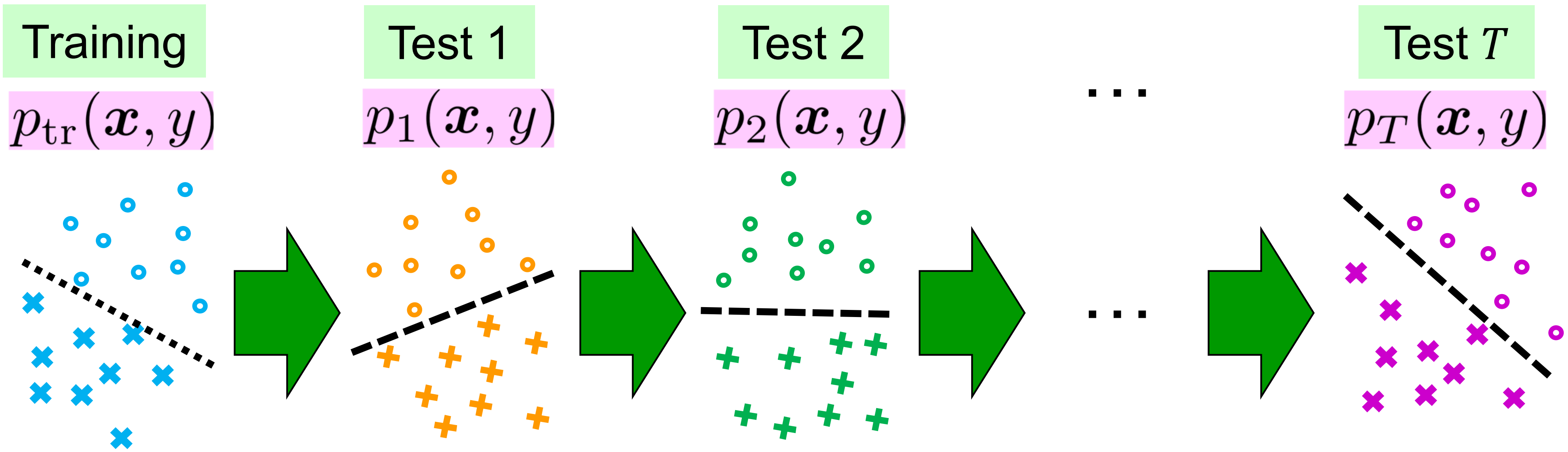
$$R_\ell(f) = \mathbb{E}_{p_{\mathrm{te}}(\boldsymbol{x}, y)}[\ell(f(\boldsymbol{x}), y)]$$

$$J_{\ell'}(w, f) = \mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x})}\left[\left(w(\boldsymbol{x}) - \frac{p_{\mathrm{te}}(\boldsymbol{x})}{p_{\mathrm{tr}}(\boldsymbol{x})}\right)^2\right] \quad \leftarrow 1^{\mathrm{st}} \text{ step}$$

$$+ (\mathbb{E}_{p_{\mathrm{tr}}(\boldsymbol{x}, y)}[w(\boldsymbol{x}) \ell'(f(\boldsymbol{x}), y)])^2 \quad \leftarrow 2^{\mathrm{nd}} \text{ step}$$

- Classic approach corresponds to 2-step minimization.

Extensions to Continuous Shifts



- **Continuous label shift:** Bai+ (NeurIPS2022)
  - Only class-prior $p_t(y)$ changes.
- **Continuous covariate shift:** Zhang+ (arXiv2023)
  - Only input density $p_t(\boldsymbol{x})$ changes.
- Without knowing the shift intensity, our methods achieve the same dynamic regret as the case with known shift intensity.

$$\mathbb{E}\left[\sum_{t=1}^{T} R_t(f_t) - \sum_{t=1}^{T} \min_{f \in \mathcal{F}} R_t(f)\right]$$

Contents

Joint Shift

- Many distribution shift works focus on a particular shift type (e.g., covariate shift):

$$p_{\mathrm{tr}}(\boldsymbol{x}) \neq p_{\mathrm{te}}(\boldsymbol{x}) \qquad p_{\mathrm{tr}}(y|\boldsymbol{x}) = p_{\mathrm{te}}(y|\boldsymbol{x})$$

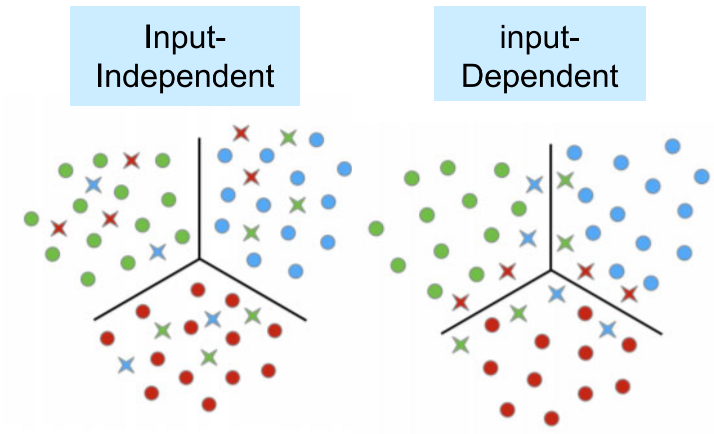
  - However, identification of the shift type is challenging.

- Label noise is also a type of distribution shift:

$$p_{\mathrm{tr}}(\bar{y}|\boldsymbol{x}) = \sum_{y} \underbrace{p(\bar{y}|y,\boldsymbol{x})}_{\text{Noise transition}} p_{\mathrm{te}}(y|\boldsymbol{x})$$

$\bar{y}$: Noisy class label

  - Nice theory for input-independent noise.
  - But input-dependent noise is hard.



- Let's consider joint shift:

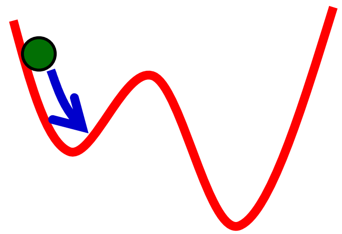
$$p_{\mathrm{tr}}(\boldsymbol{x}, y) \neq p_{\mathrm{te}}(\boldsymbol{x}, y)$$

Mini-Batch-Wise Loss Matching

Fang+ (NeurIPS2020)

- Given:
  - (Large) labeled training data: $\{(\boldsymbol{x}_i^{\mathrm{tr}}, y_i^{\mathrm{tr}})\}_{i=1}^{n_{\mathrm{tr}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x}, y)$
  - (Small) labeled test data: $\{(\boldsymbol{x}_j^{\mathrm{te}}, y_j^{\mathrm{te}})\}_{j=1}^{n_{\mathrm{te}}} \overset{\mathrm{i.i.d.}}{\sim} p_{\mathrm{te}}(\boldsymbol{x}, y)$

- We try to learn the importance weight dynamically in the mini-batch-wise manner.



$$f \leftarrow f - \eta \nabla \widehat{R}(f)$$

$\eta > 0$: step size

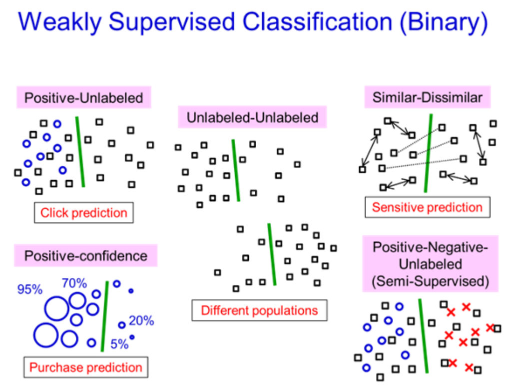
- For each mini-batch $\{(\tilde{\boldsymbol{x}}_i^{\mathrm{tr}}, \tilde{y}_i^{\mathrm{tr}})\}_{i=1}^{\tilde{n}_{\mathrm{tr}}}, \{(\tilde{\boldsymbol{x}}_j^{\mathrm{te}}, \tilde{y}_j^{\mathrm{te}})\}_{j=1}^{\tilde{n}_{\mathrm{te}}}$, importance weights are estimated by kernel mean matching for loss values:

Huang+ (NeurIPS2006)

$$\frac{1}{\tilde{n}_{\mathrm{tr}}} \sum_{i=1}^{\tilde{n}_{\mathrm{tr}}} r_i \ell(f(\tilde{\boldsymbol{x}}_i^{\mathrm{tr}}), \tilde{y}_i^{\mathrm{tr}}) \approx \frac{1}{\tilde{n}_{\mathrm{te}}} \sum_{j=1}^{\tilde{n}_{\mathrm{te}}} \ell(f(\tilde{\boldsymbol{x}}_j^{\mathrm{te}}), \tilde{y}_j^{\mathrm{te}})$$

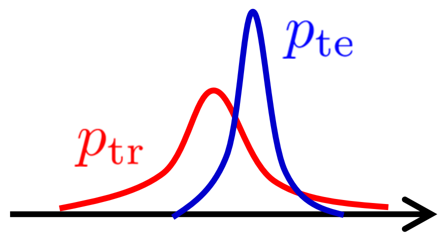
$$r_i \approx \frac{p_{\mathrm{te}}(\tilde{\boldsymbol{x}}_i^{\mathrm{tr}}, \tilde{y}_i^{\mathrm{tr}})}{p_{\mathrm{tr}}(\tilde{\boldsymbol{x}}_i^{\mathrm{tr}}, \tilde{y}_i^{\mathrm{tr}})}$$

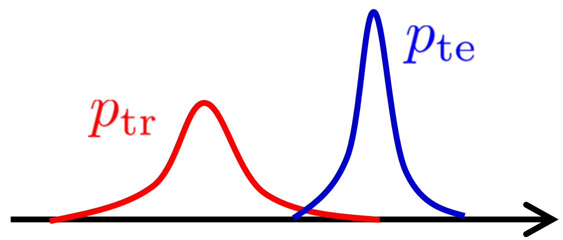
Current Challenges



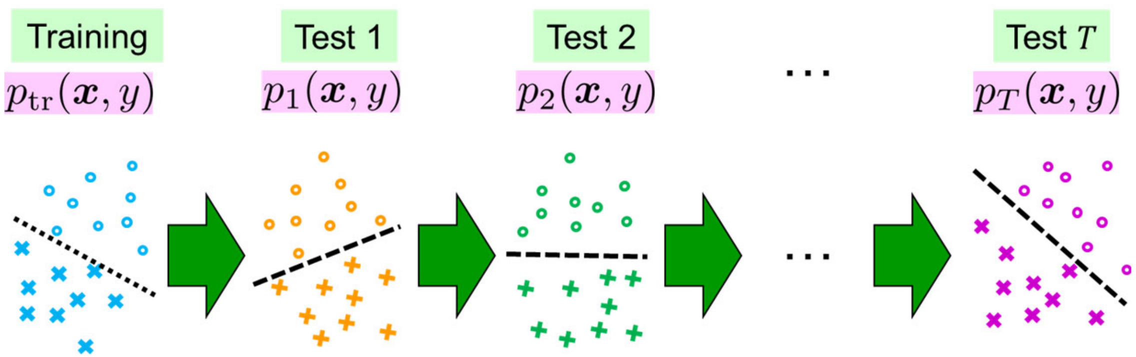
- For joint shift adaptation, requiring labeled test data is too strong.
  - Can we use weakly supervised learning?
- Importance weighting requires the test domain to be included in the training domain.
  - Can we properly handle out-of-training-domain test data?

$$\frac{p_{\mathrm{te}}(\boldsymbol{x}, y)}{p_{\mathrm{tr}}(\boldsymbol{x}, y)} < \infty$$





- Can we handle continuous joint shift?

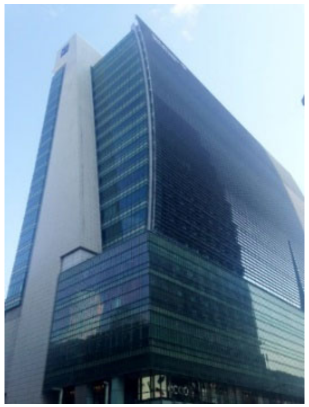


Grateful to Collaborators!

Team leader Masashi Sugiyama	Research scientist Gang Niu
Postdoctoral researcher Jingfeng Zhang	Postdoctoral researcher Jiaqi Lyu
Postdoctoral researcher Shuo Chen	Senior visiting scientist Shinichi Nakajima
Visiting scientist Futoshi Futami	Visiting scientist Florian Yger
Visiting scientist Takashi Ishida	Visiting scientist Miao Xu
Visiting scientist Takayuki Osa	Visiting scientist Bo Han
Visiting scientist Takahiro Mimori	Visiting scientist Feng Liu
Visiting scientist Lei Feng	Visiting scientist Tongliang Liu
Part-time worker I Masahiro Fujisawa	Part-time worker I Yifan Zhang

and many great interns!

- Professor
  - Masashi Sugiyama (Complexity, Computer, Information, RIKEN)
- Associate Professor
  - Naoto Yokoya (Complexity, Computer, Information, RIKEN)
- Lecturer
  - Takashi Ishida (Complexity, Computer, Information)
- Project Lecturer
  - Nobutaka Ito (Complexity)
- Professor (to Sato Lab from April 202
  - Issei Sato (Computer, Informati
- Project Assistant Professor
  - Chao-Kai Chiang (Complexity)
- Project Researcher (Postdoctoral Rese
  - Dongxian Wu (Complexity)
- Project Specialist
  - Yuko Kawashima (Complexity)
  - Soma Yokoi (Complexity)
  - Fumi Sato (Complexity)



- Doctoral Student
  - Shinji Nakadai (Computer)
  - Ryuichi Kiryo (Computer)
  - Jongyeong Lee (Computer)
  - Tianyi Zhang (Complexity)
  - Yivan Zhang (Computer)
  - Riou Charles (Computer)
  - Valliappa Chockalingam (Comput
  - Tongtong Fang (Complexity)
  - Boyo Chen (Complexity)
  - Xiaoyu Dong (Complexity)
  - Yujie Zhang (Complexity)
  - Xinqiang Cai (Complexity)
  - Jian Song (Complexity)
  - Wanshui Gan (Complexity)
  - Yuting Tang (Complexity)
  - Shintaro Nakamura (Complexity)
  - Or Raveh (Complexity)
  - Johannes Ackermann (Computer
  - Wei Wang (Complexity)
  - Hongruixuan Chen (Complexity)
  - Huanjian Zhou (Complexity)
  - Zhiyuan Zhan (Complexity)
  - Zhihao Liu (Complexity)
- Master Student
  - Hyunggyu Park (Complexity)* Sato lab.
  - Jiahuan Li (Computer)
  - Kun Yang (Complexity)
  - Xiaomou Hou (Complexity)
  - Anan Methasate (Computer)
  - Cemal Erat (Computer)
  - Kento Yamamoto (Computer)
  - Kazuki Ota (Computer)
  - Iu Yahiro (Computer)
  - Hikaru Fujita (Computer)
  - Yu Yao (Complexity)
  - Yoshifumi Nakano (Complexity)
  - Soichiro Nishimori (Complexity)
  - Ryota Ushio (Complexity)
  - Tiankui Xian (Complexity)
  - Thanawat Lodkaew (Computer)
  - Masahiro Negishi (Computer)
  - Yuto Nozaki (Computer)
  - Kanta Shimizu (Computer)
  - Zhongle Zhu (Computer)
  - Fang Liu (Computer)
  - Ming Li (Complexity)
  - Luheng Wang (Complexity)
  - Liuzhuozheng Li (Complexity)
- Research Student
  - Meike Tütken (Computer)
  - Serhii Khomenko (Information Science)
  - Artem Lubkivskyi (Information Science)