Deep Learning and Artificial Intelligence Winter School 2020 (DLAI4)

Nov. 20, 2020

# Robust Machine Learning for Reliable Deployment

Masashi Sugiyama

RIKEN Center for Advanced Intelligence Project/
The University of Tokyo

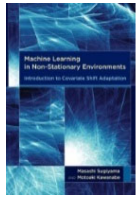
http://www.ms.k.u-tokyo.ac.jp/sugi/

# About Myself

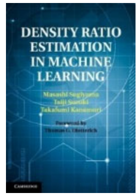
- **My jobs**:
  - Director: RIKEN AIP
  - Professor: University of Tokyo
  - Consultant: several local startups

- **Interests: Machine learning (ML)**
  - Weakly-supervised learning,
  - Robust learning,
  - Transfer learning,
  - Density ratio estimation,
  - Reinforcement learning,
  - Variational inference…
- **Academic activities:**
  - Program Chairs for NeurIPS2015, AISTATS2019, ACML2010/2020…

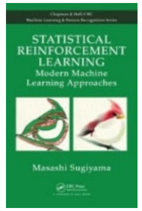
Sugiyama & Kawanabe, Machine Learning in Non-Stationary Environments, MIT Press, 2012



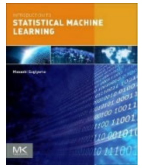
Sugiyama, Suzuki & Kanamori, Density Ratio Estimation in Machine Learning, Cambridge University Press, 2012



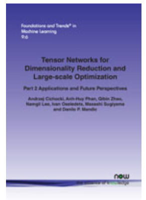

Sugiyama, Statistical Reinforcement Learning, Chapman and Hall/CRC, 2015



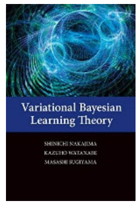

Sugiyama, Introduction to Statistical Machine Learning, Morgan Kaufmann, 2015



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

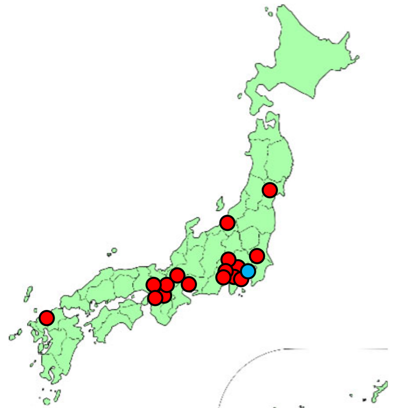


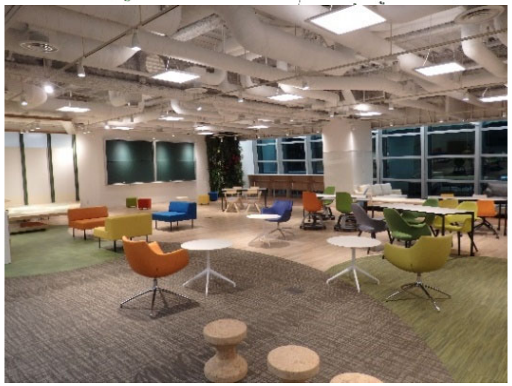
Nakajima, Watanabe & Sugiyama, Variational Bayesian Learning Theory, Cambridge University Press, 2019

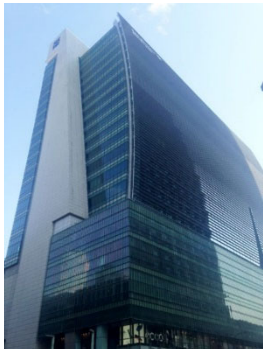


# RIKEN Center for Advanced Intelligence Project (AIP)

- 10-year national project in Japan (2016-2025):
  - Develop next-generation AI technology (learning and optimization theory, etc.)
  - Accelerate scientific research (material, cancer, stem cells, genomics, etc.)
  - Solve socially critical problems (natural disaster, elderly healthcare, etc.)
  - Study of ethical, legal and social issues of AI (ethical guideline, privacy protection, etc.)
  - Human resource development (150+ researchers, 200+ students, 150+ interns, 300+ visiting scientists, 40+ industry projects)







# Today's Topic: Robust Machine Learning

- In real-world applications, it becomes increasingly important to consider robustness:
  - Noise: sensor error, human error
  - Insufficient information: weak supervision
  - Bias: sample selection bias, changing environments
  - Attack: adversarial noise, distribution shift
- In this lecture, I will give an overview of our recent advances in robust machine learning.

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

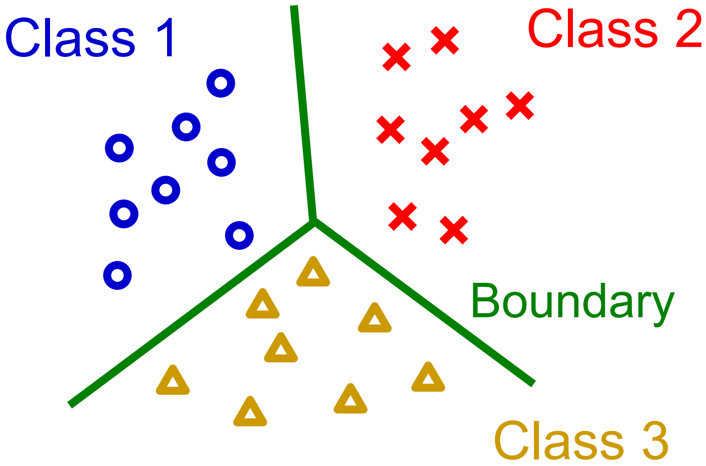
# Contents



1. Noisy label learning
2. Weakly supervised learning
3. Bias in training data
4. Noise in test input
5. Future outlook

# Ordinary Classification

- Clean training data: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$




$\boldsymbol{x} \in \mathbb{R}^d$ : Input pattern

$y \in \{1, \ldots, c\}$ : Clean class label (not necessarily separable)

- Training error minimization is statistically consistent and work well:

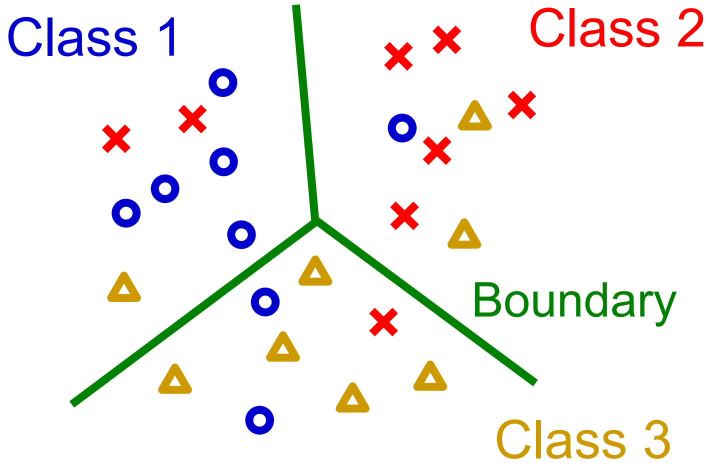
$$\frac{1}{n}\sum_{i=1}^{n} \ell\Big(y_i, \boldsymbol{g}(\boldsymbol{x}_i)\Big)$$

$\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^c$ : Classifier

$\ell(y, \boldsymbol{g}(\boldsymbol{x})) \in \mathbb{R}$ : Loss

# Noisy Classification

- Noisy training data: $\{(\boldsymbol{x}_i, \widetilde{y}_i)\}_{i=1}^{n}$




$\boldsymbol{x} \in \mathbb{R}^d$: Input pattern

$\widetilde{y} \in \{1, \ldots, c\}$: Noisy class label (clean labels are corrupted)

- Training error minimization is no longer consistent and does not work well:

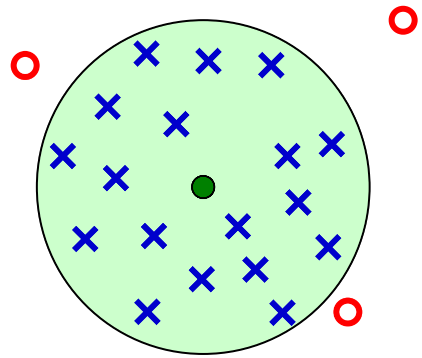
$$\frac{1}{n}\sum_{i=1}^{n} \ell\left(\widetilde{y}_i, \boldsymbol{g}(\boldsymbol{x}_i)\right)$$

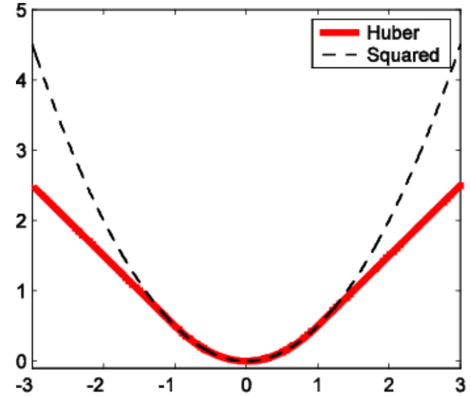
$\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^c$ : Classifier

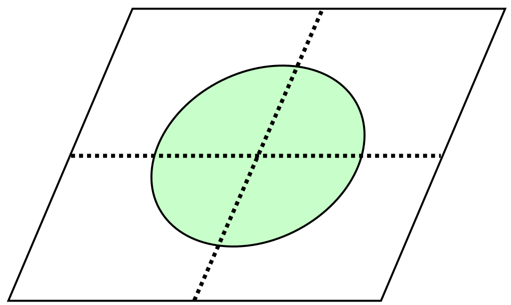
$\ell(y, \boldsymbol{g}(\boldsymbol{x})) \in \mathbb{R}$ : Loss

# Standard Approaches

- Unsupervised outlier removal:
  - Substantially difficult
- Robust loss, regularization:
  - Not robust enough
- We want to go beyond the limitations of existing approaches!
  - Noise transition correction
  - Noiseless sample selection
  - Model capacity control





# (1-1) Noise Transition Correction

- Noise transition matrix $\boldsymbol{T}$:
  - Flipping probability from $y$ to $\widetilde{y}$.

$$\boldsymbol{T}^\top = \begin{array}{c|ccc} & & y & \\ \hline \widetilde{y} & 1 & 0.1 & 0.5 \\ & 0 & 0.8 & 0.5 \\ & 0 & 0.1 & 0 \end{array}$$

- Major approaches: Patrini et al. (CVPR2017)
  - Loss correction by $\boldsymbol{T}^{-1}$ to eliminate noise.
  - Classifier correction by $\boldsymbol{T}^\top$ to simulate noise.
- We want to estimate $\boldsymbol{T}$ only from noisy data:
  - Use human cognition as a “mask” for $\boldsymbol{T}$.

    Han, Yao, Niu, Zhou, Tsang, Zhang & Sugiyama (NeurIPS2018)
  - Learn $\boldsymbol{T}$ and a classifier simultaneously.

    Xia, Liu, Wang, Han, Gong, Niu & Sugiyama (NeurIPS2019)
  - Decompose $\boldsymbol{T}$ into simpler components.

    Yao, Liu, Han, Gong, Deng, Niu, Sugiyama & Tao (NeurIPS2020)
  - Extension to input-dependent noise $\boldsymbol{T}(\boldsymbol{x})$.

    Xia, Liu, Han, Wang, Gong, Liu, Niu, Tao & Sugiyama (NeurIPS2020)

# (1-2) Co-teaching

## Memorization of neural nets:

Arpit et al. (ICML2017)
Zhang et al. (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.

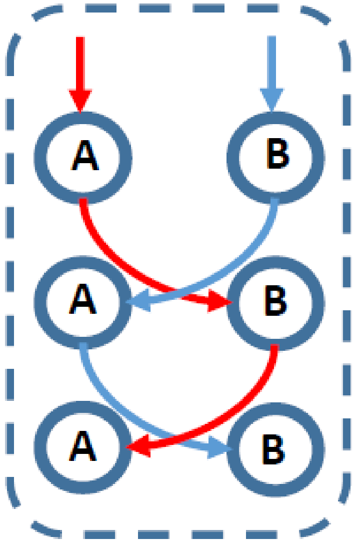
  Han, Yao, Yu, Niu, Xu, Hu, Tsang & Sugiyama (NeurIPS2018)

- Teach only disagreed data.

  Yu, Han, Yao, Niu, Tsang & Sugiyama (ICML2019)

- Gradient ascent for large-loss data.

  Han, Niu, Yu, Yao, Xu, Tsang & Sugiyama (ICML2020)



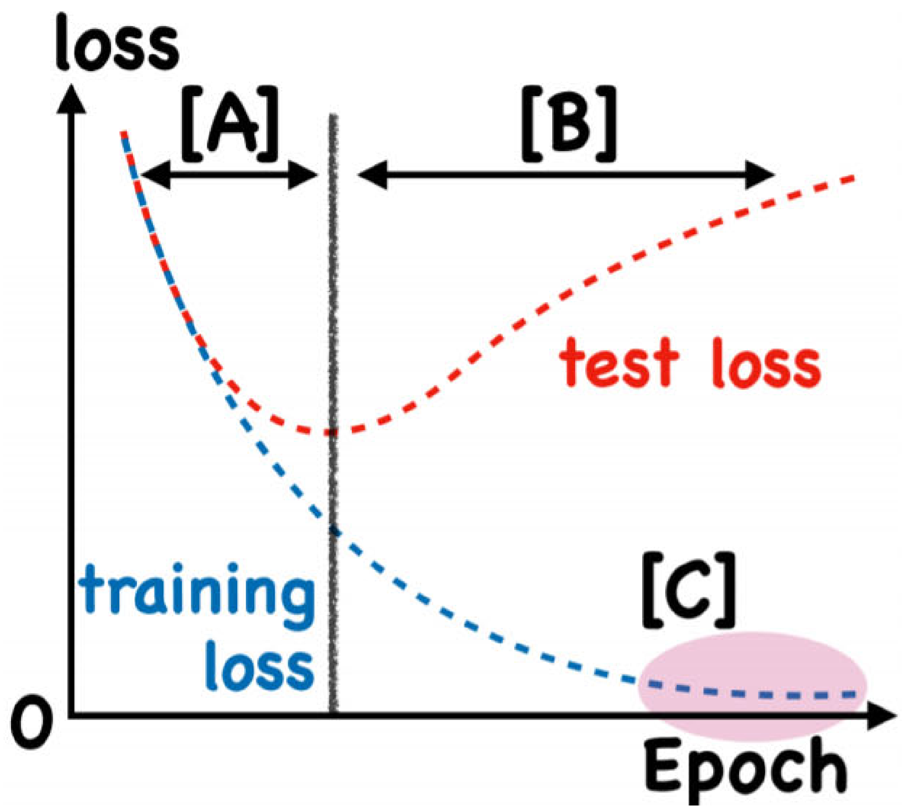

## No theory but very robust in experiments:

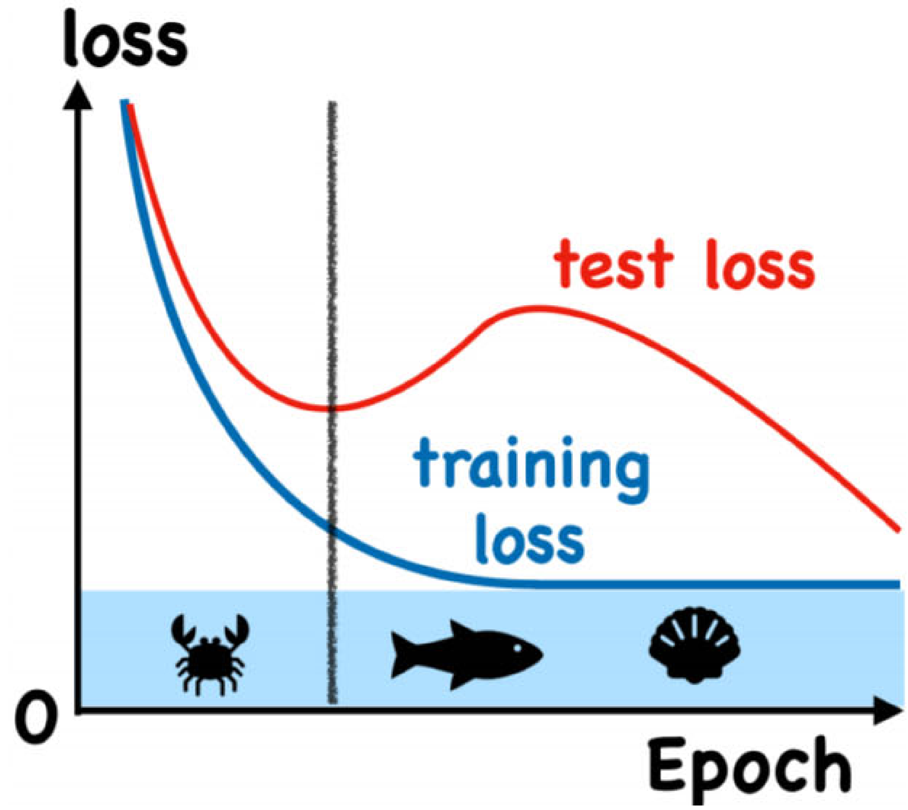
- Works well even if 50% labels are randomly flipped.

# (1-3) Flooding

- Neural nets tend to overfit.
- "Flooding" the training error prevents overfitting.
  - It induces double descent? $|R(f)-b|+b$

Ishida, Yamane, Sakai, Niu & Sugiyama (ICML2020)

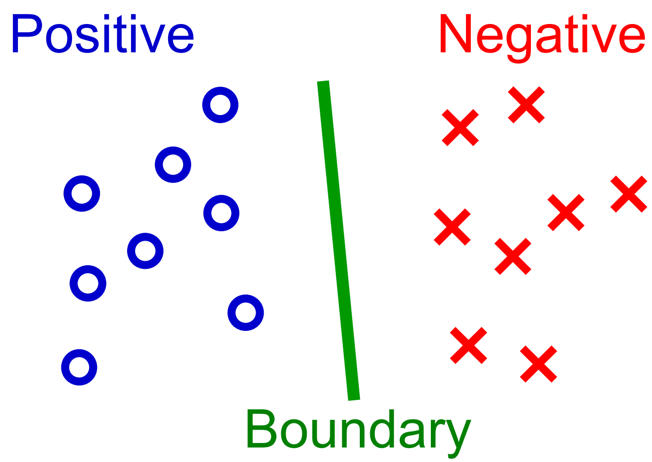


# Contents

# Weakly Supervised Learning

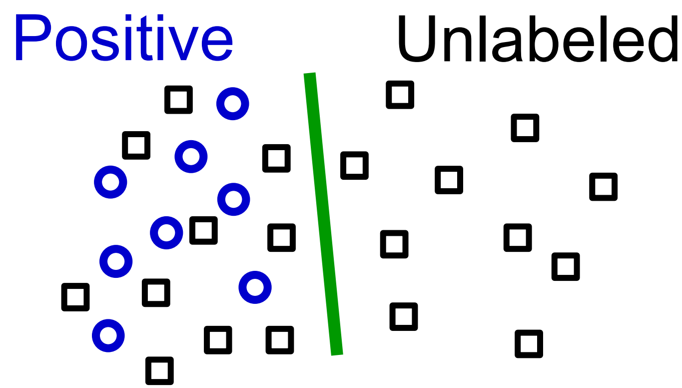
- Ordinary supervised learning requires fully labeled data (input-output pairs).
- But collecting fully labeled data can be expensive in practice.
- Can we utilize "weakly" labeled data?
  - No negative data
  - Positive confidence data
  - Similar/dissimilar data
  - Complementary data
  - Partial-label data




P: Positive, N: Negative, U: Unlabeled

# (2-1) PU Classification

- Only positive and unlabeled data is available; negative data is completely missing:
  - Click vs. non-click
  - Friend vs. non-friend




- We want to minimize the risk of classifier $f$ only from PU data:

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

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

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

- But N-risk cannot be estimated directly.

# Key Trick

du Plessis, Niu & Sugiyama (NIPS2014, ICML2015)
Niu, du Plessis, Sakai, Ma & Sugiyama (NIPS2016)
Kiryo, du Plessis, Niu & Sugiyama (NIPS2017)
Hsieh, Niu & Sugiyama (ICML2019)

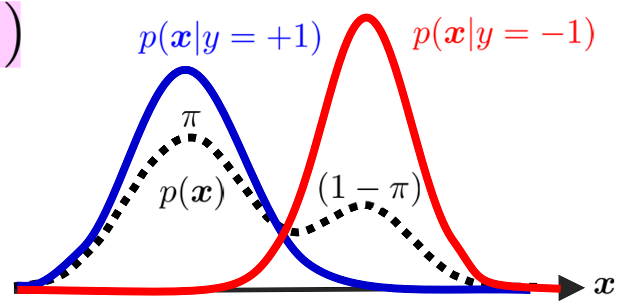
Risk for P data — Risk for N data

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

- Use "U-density is mixture of P- and N-densities":

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

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




  - Then

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

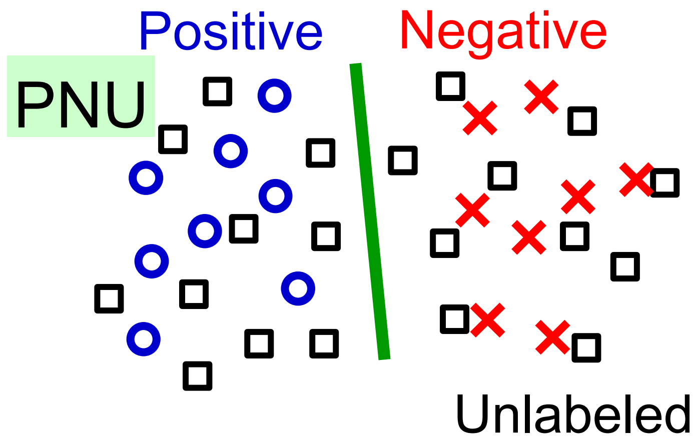
  - Empirical risk minimization (ERM) is possible from PU data, just by replacing expectations by sample averages!

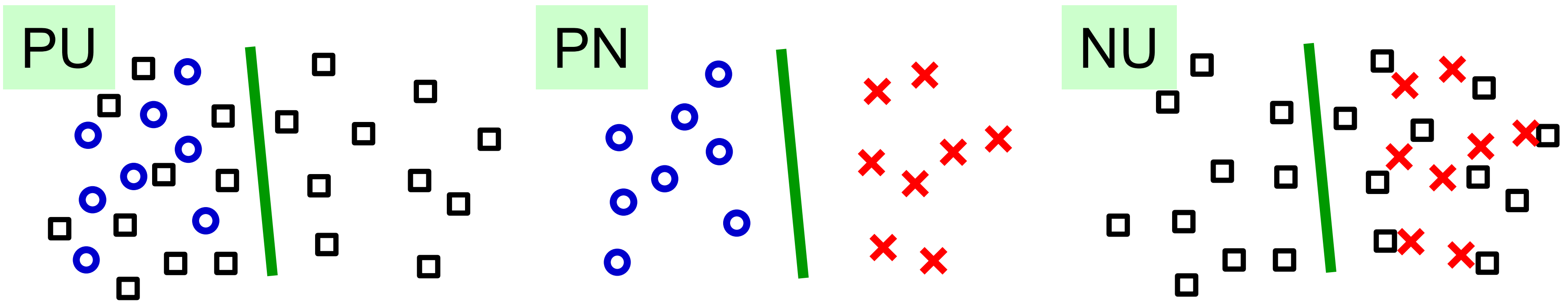
$$R(\widehat{f}_{\mathrm{PU}}) - R(f^*) \leq C(\delta)\left(\frac{2\pi}{\sqrt{n_{\mathrm{P}}}} + \frac{1}{\sqrt{n_{\mathrm{U}}}}\right)$$

# (2-2) PNU Classification (Semi-Supervised Classification)

Sakai, du Plessis, Niu & Sugiyama (ICML2017)
Sakai, Niu & Sugiyama (MLJ2018)

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




$$R_{0/1}(f) \leq 2\widehat{R}^{\gamma}_{\mathrm{PN+PU}}(f) + \mathcal{O}(1/\sqrt{n_{\mathrm{P}}} + 1/\sqrt{n_{\mathrm{N}}} + 1/\sqrt{n_{\mathrm{U}}})$$

# (2-3) Pconf Classification

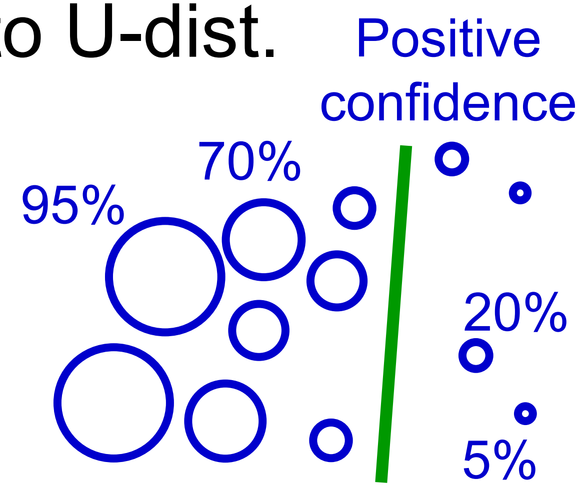
Ishida, Niu & Sugiyama (NeurIPS2018)

- Only P data is available, even not U data:
  - Data from rival companies cannot be obtained.
  - Only positive results are reported (publication bias).
- "Only-P learning" is unsupervised.
- From positive-confidence data, ERM is possible!
  - Augment r-Pconf samples to (1-r)-Nconf samples.
  - Importance sampling from P-dist. to U-dist.

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

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

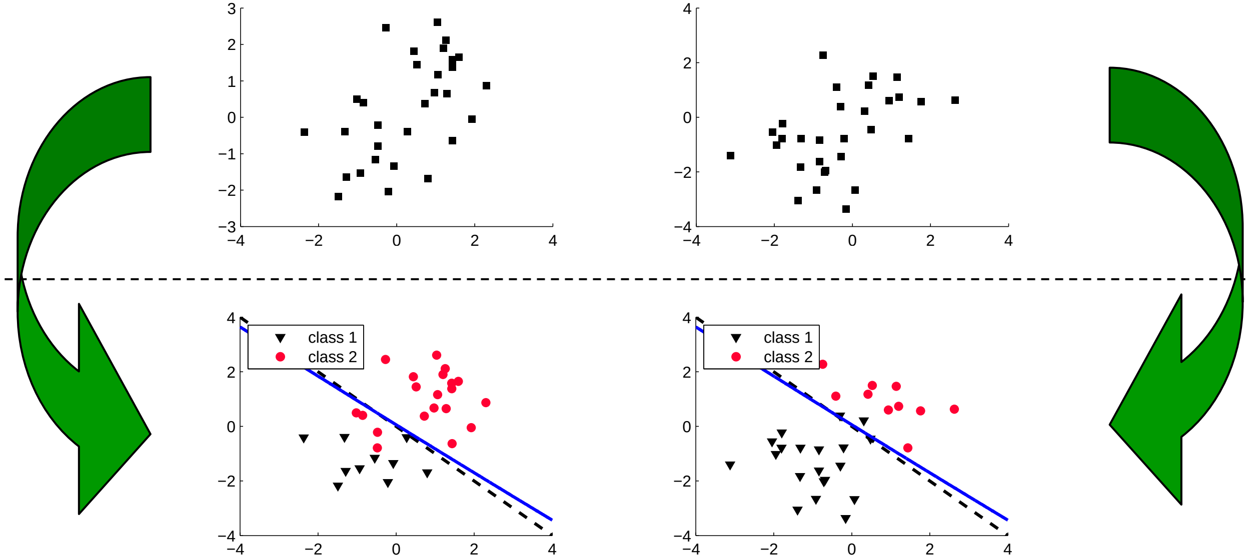
$$R\left(f^*\right) - R\left(\widehat{f}\right) = \mathcal{O}_p\left(1/\sqrt{n}\right)$$



# (2-4) UU Classification

du Plessis, Niu & Sugiyama (TAAI2013)
Lu, Niu, Menon & Sugiyama (ICLR2019)
Charoenphakdee, Lee & Sugiyama (ICML2019)
Lu, Zhang, Niu & Sugiyama (AISTATS2020)

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




- In PU, we regarded U as noisy N.
- In UU, we use noisy P and noisy N!

$\mathcal{O}_p\left(1/\sqrt{n}\right)$

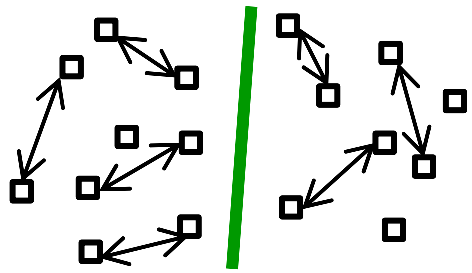
# (2-5) SU Classification



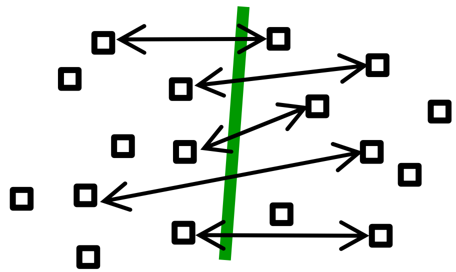
Bao, Niu & Sugiyama (ICML2018)

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

$1/\sqrt{n}$



  - Decoupling S-pairs results in UU classification!
- Learning from dissimilar data pairs is also possible.



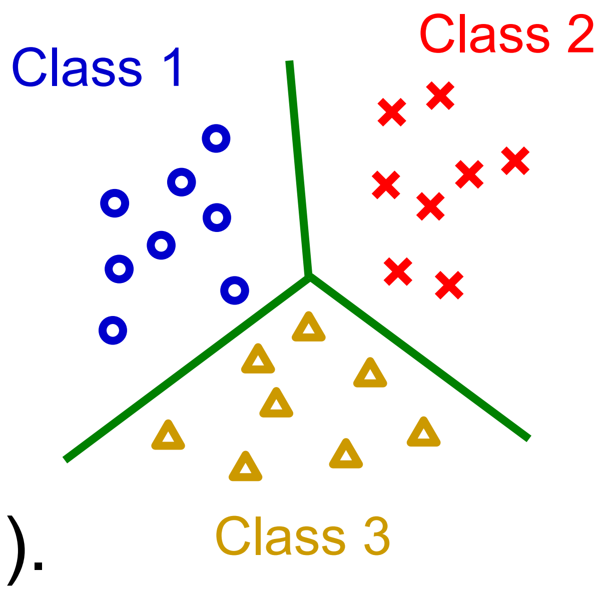
  - SDU classification is also possible.

Shimada, Bao, Sato & Sugiyama (NeCo, to appear)
Dan, Bao & Sugiyama (arXiv2020)

# (2-6) Complementary Classification

Ishida, Niu & Sugiyama (NIPS2017)
Ishida, Niu, Menon & Sugiyama (ICML2019)
Feng, Kaneko, Han, Niu, An & Sugiyama (ICML2020)
Chou, Niu, Lin & Sugiyama (ICML2020)

- Labeling patterns in multi-class problems:
  - Selecting the collect class from a long class list is extremely painful.




- Complementary labels:
  - Specify a class that a pattern does not belong to ("not class 1").
  - This is much easier and faster to collect!
- From complementary labels, classifiers are trainable by ERM!
  - Noisy labels with uniform transition to other classes.

$1/\sqrt{n}$

# 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) |
|---|---|

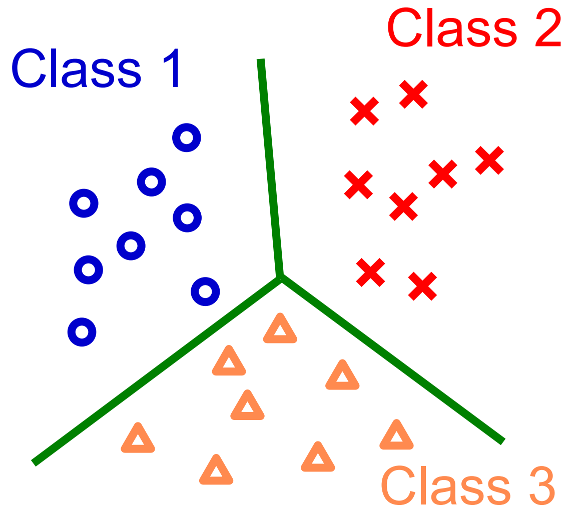
- Use both of ordinary and complementary labels!

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

# (2-7) Partial-Label Classification



- Partial label: Nguyen and Caruana (KDD2008)
  a subset of labels containing the true one
  - "Either 1 or 2"
  - Cheaper than ordinary labels




- From partial labels, classifiers are trainable by ERM!
  - Complementary label is a special case of partial label.

Lv, Xu, Feng, Niu, Geng & Sugiyama (ICML2020)
Feng, Lv, Han, Xu, Niu, Geng, An & Sugiyama (NeurIPS2020)

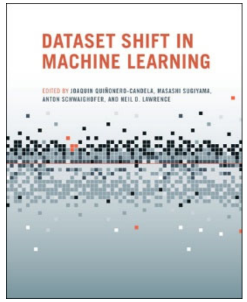
# Contents



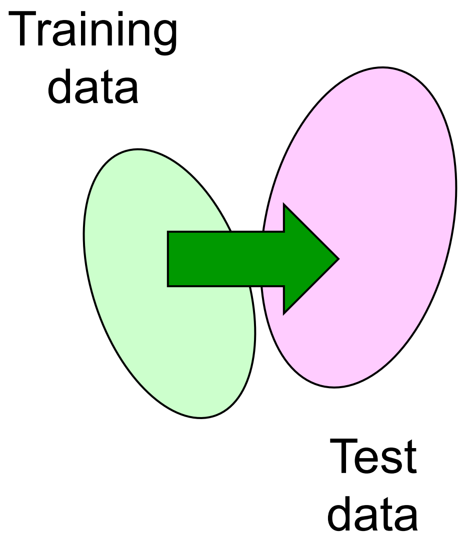
1. Noisy label learning
2. Weakly supervised learning
3. Bias in training data
4. Noise in test input
5. Future outlook

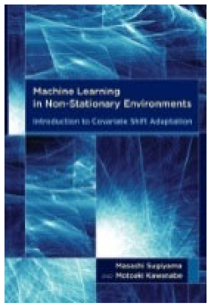
# Bias in Training Data

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




- Training and test data often have different distributions, due to
  - changing environments,
  - sample selection bias.
- Transfer learning/domain adaptation:
  - Match the distributions so that training data resemble test data.




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

# Unsupervised Transfer Learning



- Given training input-output and test input, match the training and test distributions:
  - Better discrepancy measures for matching:

    Kuroki, Charoenphakdee, Bao, Honda, Sato & Sugiyama (AAAI2019)
    Lee, Charoenphakdee, Kuroki & Sugiyama (arXiv2019)

  - Handling noisy labels in the source domain:

    Liu, Lu, Han, Niu, Zhang & Sugiyama (arXiv2019)

  - No/incomplete unlabeled data from the test domain:

    Ishii, Takenouchi & Sugiyama (ACML2019)
    Ishii, Takenouchi & Sugiyama (WACV2020)

  - Transferring data generation mechanism:

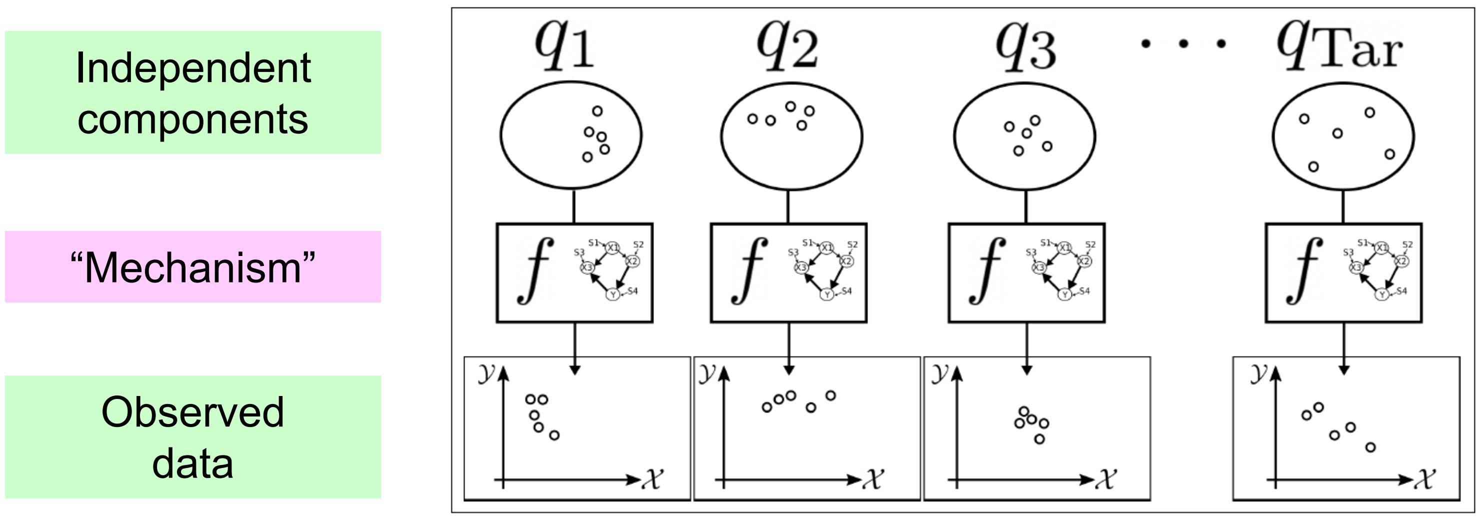
    Teshima, Sato & Sugiyama (ICML2020)
    Teshima, Ishikawa, Tojo, Oono, Ikeda & Sugiyama (NeurIPS2020)

  - Simultaneous learning of a classifier and importance weights:

    Zhang, Yamane, Lu & Sugiyama (ACML2020)
    Fang, Lu, Niu & Sugiyama (NeurIPS2020)

# (3-1) Mechanism Transfer

- Is transfer learning possible when data distributions are seemingly very different?
- Yes, if data generation mechanisms are shared:
  - Use invertible neural networks (INNs) to invert the data generation mechanism. Teshima, Sato & Sugiyama (ICML2020)
  - INNs are universal approximators. Teshima, Ishikawa, Tojo, Oono, Ikeda & Sugiyama (NeurIPS2020)

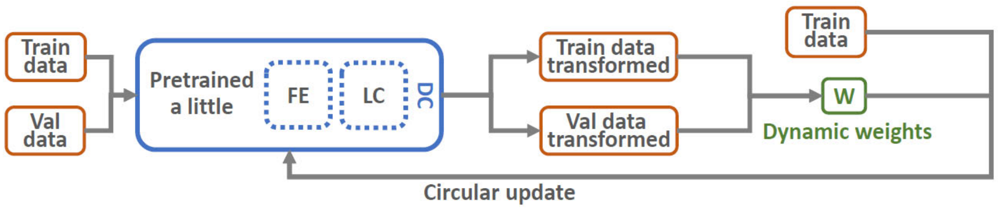


# (3-2) One-Step Adaptation



- Standard approach: 2 steps
  - Weight estimation: $\min_{w} D(w, p_{\mathrm{te}}/p_{\mathrm{tr}})$
  - Weighted classifier training: $\min_{f} \mathbb{E}_{p_{\mathrm{tr}}}[w(x,y)\ell(f(x),y)]$
- Proposed methods: 1 step
  - With a common feature extractor for $w$ and $f$, learn them dynamically in mini-batch training.

Fang, Lu, Niu & Sugiyama (NeurIPS2020)




  - Minimize an upper bound of the risk w.r.t. $w$ and $f$ under covariate shift $p_{\mathrm{tr}}(y|x) = p_{\mathrm{te}}(y|x)$:

Zhang, Yamane, Lu & Sugiyama (ACML2020)

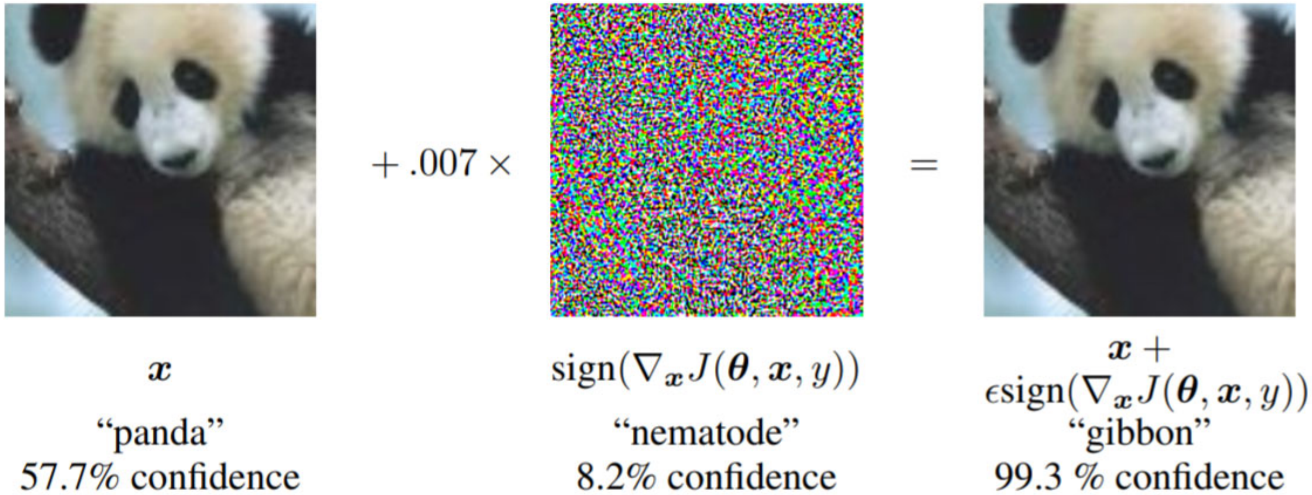
$$\min_{w,f} J(w,f) \quad J(w,f) \geq R^2(f)$$

# Contents

1. Noisy label learning
2. Weakly supervised learning
3. Bias in training data
4. Noise in test input
5. Future outlook

# Noise in Test Input

- Neural nets are vulnerable to small perturbations in test input. Goodfellow et al. (ICLR2015)




- We want to be robust to such perturbations:
  - Robust to adversarial distribution shift.
  - "Friendly" adversarial training.
  - Defense to pointwise adversarial attack.
  - Rejection of adversarial data.

# (4-1) Distributionally Robust Learning



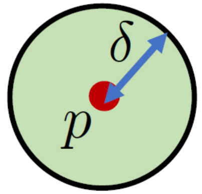
- Consider the worst-case test distribution when only training input-output is given:

$$\boxed{\min_{\theta} \sup_{q \in \mathcal{Q}_p} \mathbb{E}_{q(x,y)}[\ell(g_\theta(x), y)]}$$

$$\mathcal{Q}_p = \{q \mid \mathrm{D}_f(q \| p) \leq \delta\}$$

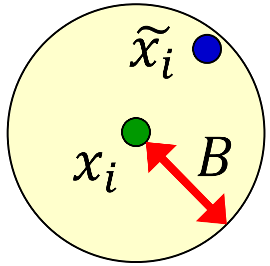
**"f-divergence ball"**

[Bagnell 2005, Ben-Tal+ 2013, Namkoong+ 2016, 2017]




- However, a naïve minimax approach does not work well:
  - Proved to be non-robust for classification.
    Hu, Niu, Sato & Sugiyama (ICML2018)
  - Elucidated the condition for loss calibration.
    Bao, Scott & Sugiyama (COLT2020)
  - New formulation for being not too conservative.
    Zhang, Xu, Han, Niu, Cui, Sugiyama & Kankanhalli (ICML2020)

# (4-2) "Friendly" Adversarial Training




## Adversarial training:

- Consider the worst test input.

$$\min_f \frac{1}{\mathrm{n}} \sum_{i=1}^{n} \ell(f(\tilde{x}_i), y_i)$$

$$\tilde{x}_i = \mathop{\arg\max}_{\tilde{x} \in B(x_i)} \ell(f(\tilde{x}), y_i)$$

- However, minimax training is too conservative.

## "Friendly" adversarial training:

Zhang, Xu, Han, Niu, Cui, Sugiyama & Kankanhalli (ICML2020)

- Among adversarial inputs, consider the one with certain margin $\rho$.

$$\tilde{x}_i = \mathop{\arg\min}_{\tilde{x} \in B(x_i)} \ell(f(\tilde{x}), y_i)$$
$$\text{s.t. } \ell(f(\tilde{x}), y_i) - \min_y \ell(f(\tilde{x}), y) \geq \rho$$

- Taking into account "geometry" can further improve the robustness.

Zhang, Zhu, Niu, Han, Sugiyama & Kankanhalli (arXiv2020)

# (4-3) Defense to Pointwise Attack



- **Stabilize output of the neural net:**

$$\forall \epsilon, \left( \|\epsilon\|_2 < c \;\Rightarrow\; t_X = \underset{i}{\operatorname{argmax}} \{F(X+\epsilon)_i\} \right)$$

- **Lipchitz-margin training:** Tsuzuku, Sato & Sugiyama (NeurIPS2018)
  - Compute the Lipchitz constant for each layer and for the entire network:

$$\|F(X) - F(X+\epsilon)\|_2 \leq L_F \|\epsilon\|_2$$

  - Train the neural net to have large prediction margins:

$$\forall i \neq t_X, (F_{t_X} \geq F_i + \sqrt{2} c L_F)$$

  - Robustness is theoretically guaranteed.

# (4-4) Classification with Reject Option 

Ni, Charoenphakdee, Honda & Sugiyama (NeurIPS2019)

- In severe applications, better to reject difficult test inputs and ask human to predict instead.
- Approach 1: Train the classifier and rejector
  - Existing methods only focus on binary problems.
  - We proved this approach does not converge to the optimal solution generally in multi-class cases.
- Approach 2: Reject low-confidence prediction
  - Existing methods have limitation in loss functions (e.g., logistic loss), resulting in weak performance.
  - New rejection criteria for general losses with theoretical convergence guarantee.

# Contents

1. Noisy label learning
2. Weakly supervised learning
3. Bias in training data
4. Noise in test input
5. Future outlook

# Summary

- Nowadays, ML systems are deployed in various societal problems, where reliability is extremely important.
- We explored robustness to different factors:
  - Noise: sensor error, human error
  - Insufficient information: weak supervision
  - Bias: sample selection bias, changing environments
  - Attack: adversarial noise, distribution shift

# Challenges in Reliable ML

- Reliable ML in expectable situations:
  - Model the corruption process explicitly and correct the solution.
- Reliable ML in unexpected situations:
  - Consider worst-case robustness,
  - Include human support.
- Exploring somewhere in the middle would be practically useful and important.
  - Partial knowledge of the corruption process.

# Challenges in Reliable ML



- In reliable ML research, the choice of performance metrics is crucial.
  - Simply improving the accuracy is not the goal.
- Since humans use ML systems, performance metrics should reflect human cognitive bias.
  - Ex: in image evaluation, MSE is not natural, but we care edges, texture, faces, etc.
- “Designing” appropriate performance metrics is an important challenge.

# Past and Future of AI Research

## Logical AI

- 1960's: Inference and search
- 1980's: Expert systems and knowledge bases

## Neuro-inspired AI

- 1960's: Single-layer perceptrons
- 1990's: Multi-layer perceptrons

## Statistical ML based AI

- 2000's: Frequentist statistics, convex optimization, Bayesian statistics
- 2010's: Deep learning

## Future AI

Human-like AI? Human-inclusive AI?