Gatsby-Kakenhi Joint Workshop on AI and Neuroscience, London

May 11-12, 2017

Classification from Weak Supervision

Masashi Sugiyama

Director, RIKEN Center for Advanced Intelligence Project (AIP)

Professor, The University of Tokyo

What Is My Talk about?

- Machine learning from big data is successful.
- However, there are various applications where massive labeled data is not available.
- In this talk, I will introduce our recent advances in classification from weak supervision .

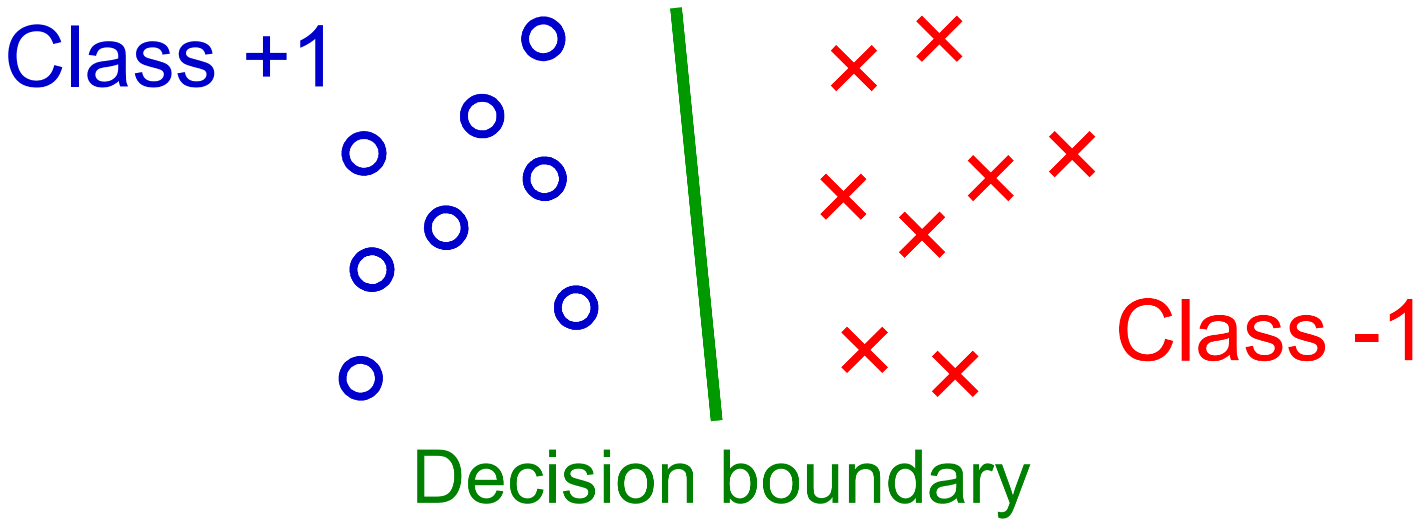
Organization

1. Classification of classification
2. PU classification
3. PNU (=PU+PN) classification
4. UU classification

Supervised Classification

- Binary classification from labeled samples:

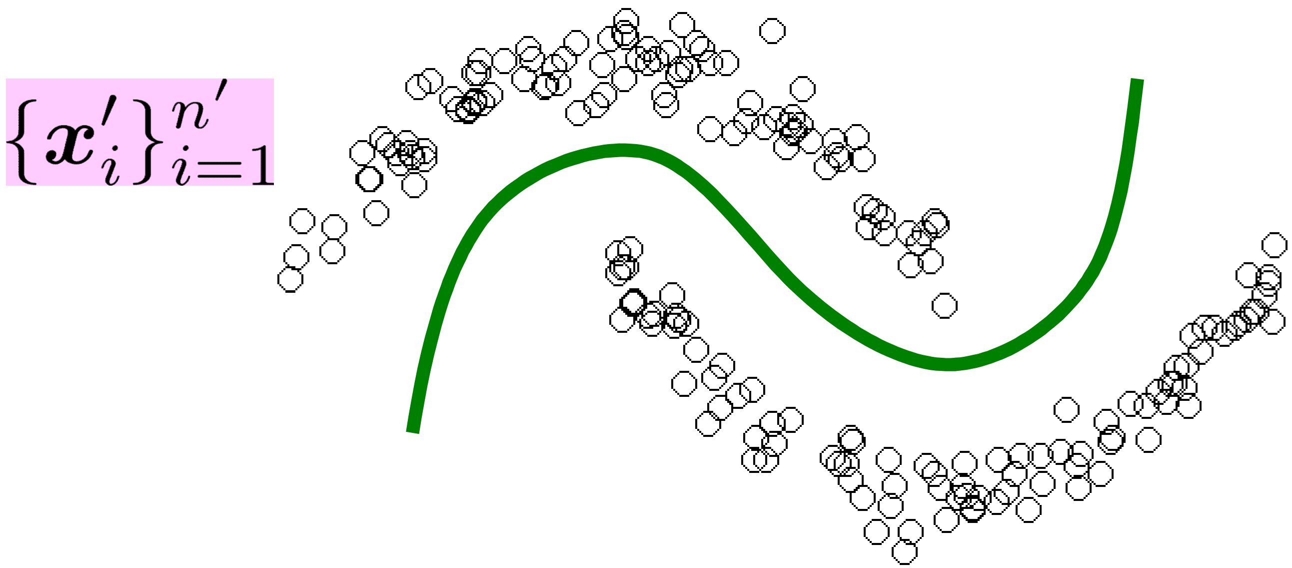
$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ $\quad \boldsymbol{x} \in \mathbb{R}^d \quad y \in \{+1, -1\}$



- A large number of labeled samples yield better classification performance.
  - Optimal convergence rate: $\mathcal{O}\left(n^{-1/2}\right)$

Unsupervised Classification

- Since collecting labeled samples is costly, let’s learn a classifier from unlabeled data.

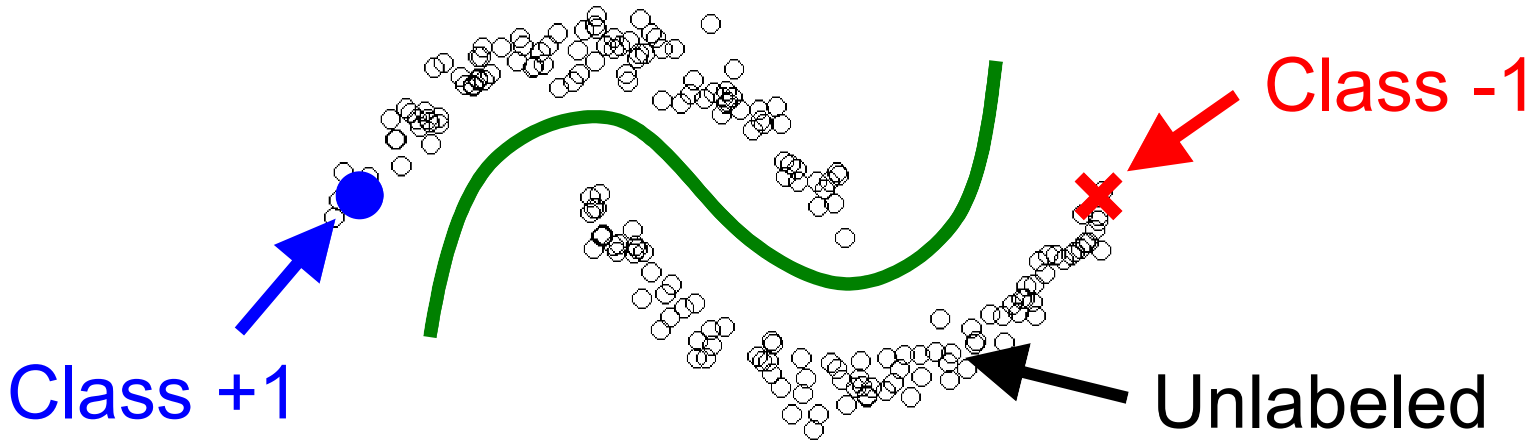


- This is equivalent to clustering.
- To justify this, need the assumption that each cluster corresponds to each class.
  - This is rarely satisfied in practice.

Semi-Supervised Classification

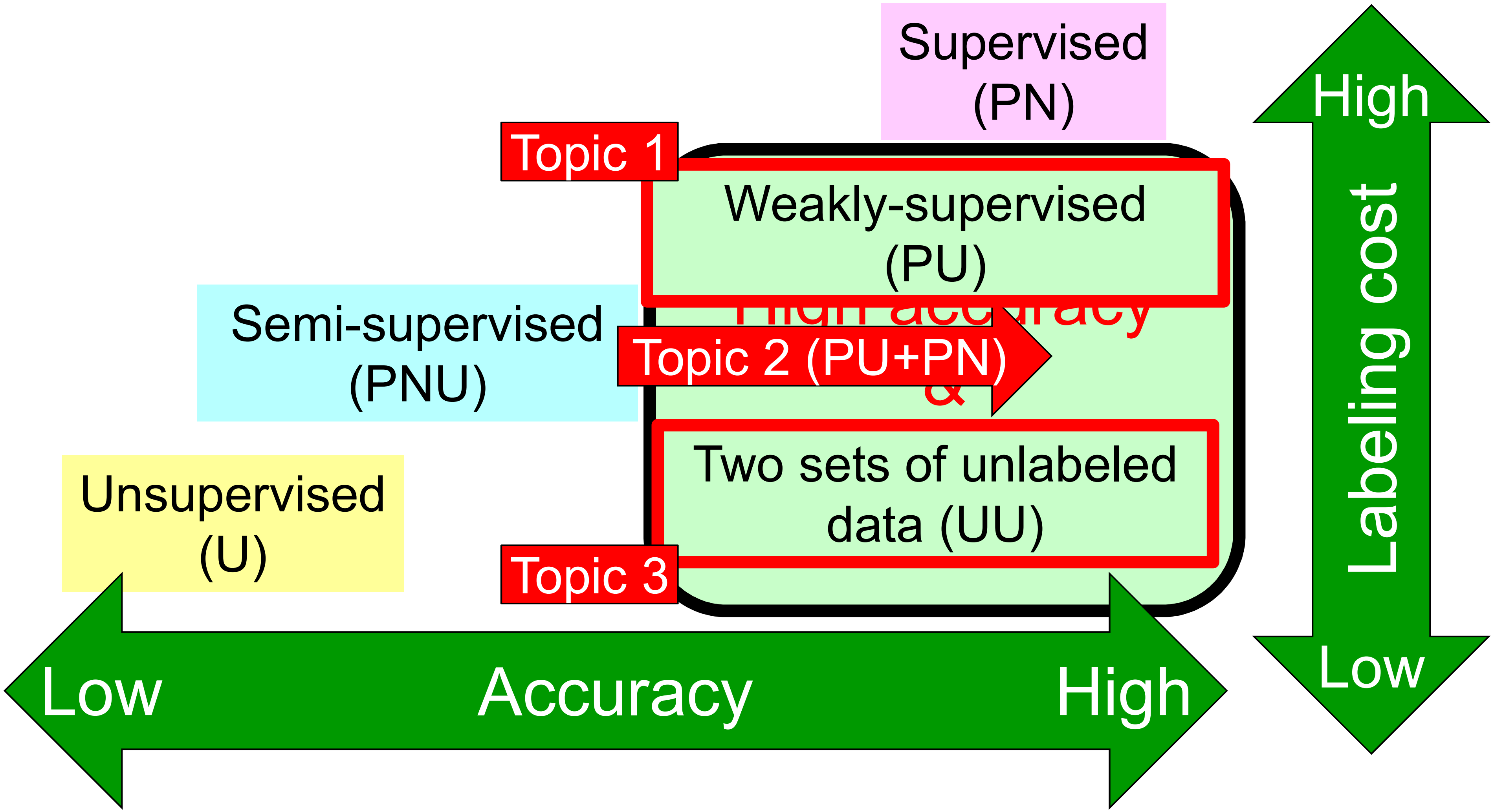
Chapelle, Schölkopf & Zien (MIT Press 2006) and many

- Use a large number of unlabeled samples and a small number of labeled samples:
- Find a decision boundary along cluster structure induced by unlabeled samples.
  - Not that different from unsupervised classification.



Classification of Classification

- Choose an appropriate formulation depending on the cost requirement.



Organization

1. Classification of classification
2. PU classification
3. PNU (=PU+PN) classification
4. UU classification

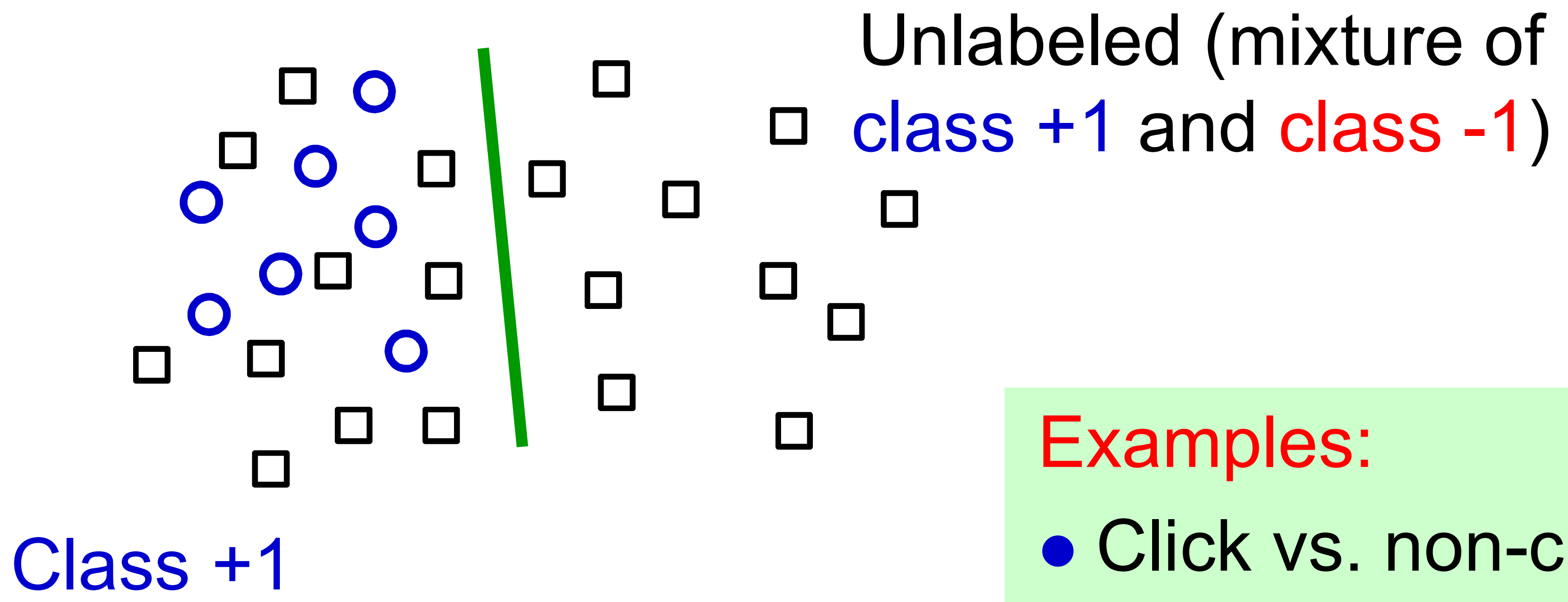
PU Classification: Setup

- **Given:** Positive and unlabeled samples

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

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

- **Goal:** Obtain a PN classifier



Examples:
- Click vs. non-click
- Friend vs. non-friend

PU Classification

- Classification risk: $R(f) = \int \ell\big(yf(\boldsymbol{x})\big)p(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}$
- Equivalent expression with PN data:

$$R(f) = \pi \int \ell\big(f(\boldsymbol{x})\big)p(\boldsymbol{x}|y=+1)\mathrm{d}\boldsymbol{x}$$

False negative rate
(P is misclassified as N)

$$+(1-\pi)\int \ell\big(-f(\boldsymbol{x})\big)p(\boldsymbol{x}|y=-1)\mathrm{d}\boldsymbol{x}$$

False positive rate
(N is misclassified as P)

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

du Plessis, Niu & Sugiyama (IEICE2014, MLj2017)

- Since no N data is available in PU setting, false positive rate cannot be estimated.

PU Classification

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

$$R(f) = \pi \int \ell\Big(f(\boldsymbol{x})\Big) p(\boldsymbol{x}|y=+1)\mathrm{d}\boldsymbol{x} + (1-\pi) \int \ell\Big(-f(\boldsymbol{x})\Big) p(\boldsymbol{x}|y=-1)\mathrm{d}\boldsymbol{x}$$

- U is a mixture of P and N:

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

  - N-risk can be estimated from PU data.

- Equivalent expression of risk without N data:

$$R(f) = \pi \int \widetilde{\ell}\Big(f(\boldsymbol{x})\Big) p(\boldsymbol{x}|y=+1)\mathrm{d}\boldsymbol{x}$$

loss function for P data

$$\widetilde{\ell}(m) = \ell(m) - \ell(-m)$$

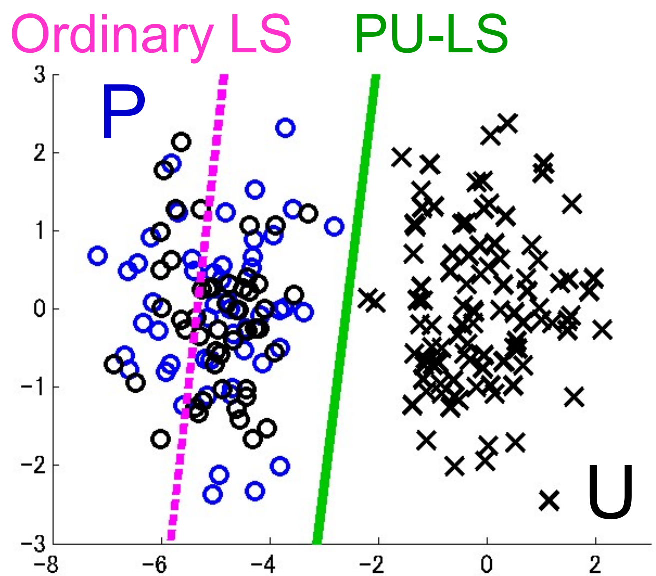
$$+ \int \ell\Big(-f(\boldsymbol{x})\Big) p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

loss function for U data

  - Unbiased estimation is possible only from P and U.

Implementation in MATLAB®

- Essentially 1 line for linear least-squares!




```
%Data generation
n=50; m=150; p=50;
x=randn(n+m,2);
x(1:n+p,1)=x(1:n+p,1)-5;
x(:,3)=1; u=x(n+1:end,:);
y=[ones(n+p,1); -ones(m-p,1)];
figure(1); z=[ones(n,1); zeros(m,1)]
plot(x(y==1&z==1,1),x(y==1&z==1,2),'bo');
plot(x(y==1&z==0,1),x(y==1&z==0,2),'ko');
plot(x(y==-1,1),x(y==-1,2),'kx');
% Computing the solution
t=(u'*u/n+0.1*eye(3))\(2*p/m*mean(x(1:n,:))-mean(u))';
plot([-10 10],-(t(3)+[-10 10]*t(1))/t(2),'k-');
```

PU for Deep Networks

Kiryo, Niu, du Plessis & Sugiyama (arXiv2017)

- Population false negative rate is non-negative:

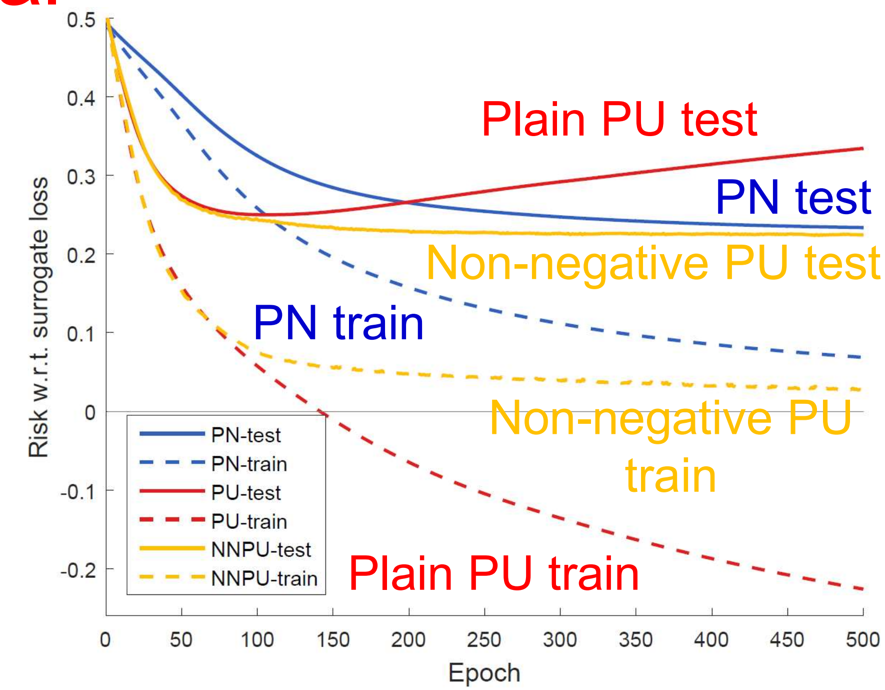
$$\int \ell\big(-f(\boldsymbol{x})\big)(1-\pi)p(\boldsymbol{x}|y=-1)\mathrm{d}\boldsymbol{x}$$

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

$$= \int \ell\big(-f(\boldsymbol{x})\big)\Big(p(\boldsymbol{x}) - \pi p(\boldsymbol{x}|y=+1)\Big)\mathrm{d}\boldsymbol{x} \geq 0$$

- However, its PU empirical approximation can be negative (in particular, for flexible deep nets).
- We impose it to be non-negative through back-prop training:

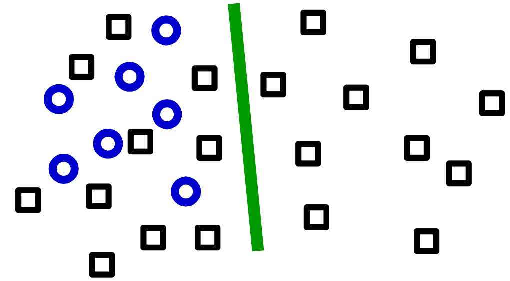
$$\max\{0, \widehat{p}(\boldsymbol{x}) - \widehat{\pi}\widehat{p}(\boldsymbol{x}|y=+1)\}$$



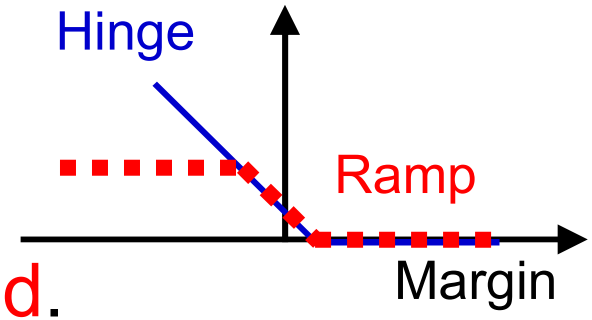
PU Classification: Summary

du Plessis, Niu & Sugiyama (NIPS2014, ICML2015)
Niu, du Plessis, Sakai, Ma & Sugiyama (NIPS2016)
Kiryo, Niu, du Plessis & Sugiyama (ArXiv2017)

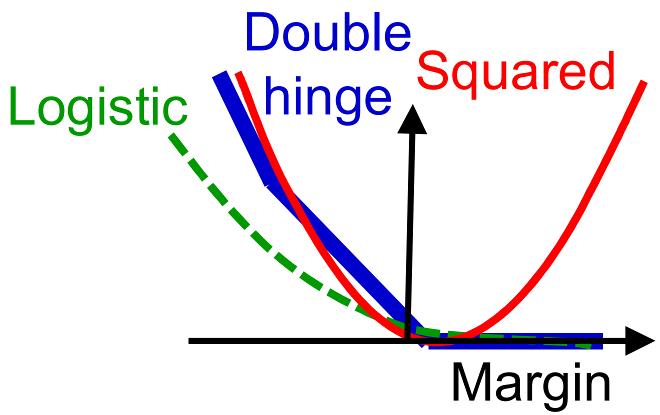
- Just separating P and U is biased.



- Use composite loss
  $\widetilde{\ell}(m) = \ell(m) - \ell(-m)$ for P data.
- If $\ell(m) + \ell(-m) = \mathrm{Const.}$,
  same loss for P and U data.
  - Optimal convergence rate achieved.



- If $\widetilde{\ell}(m) = am + b$,
  objective is convex.
- For deep nets, roundup the empirical false negative rate.

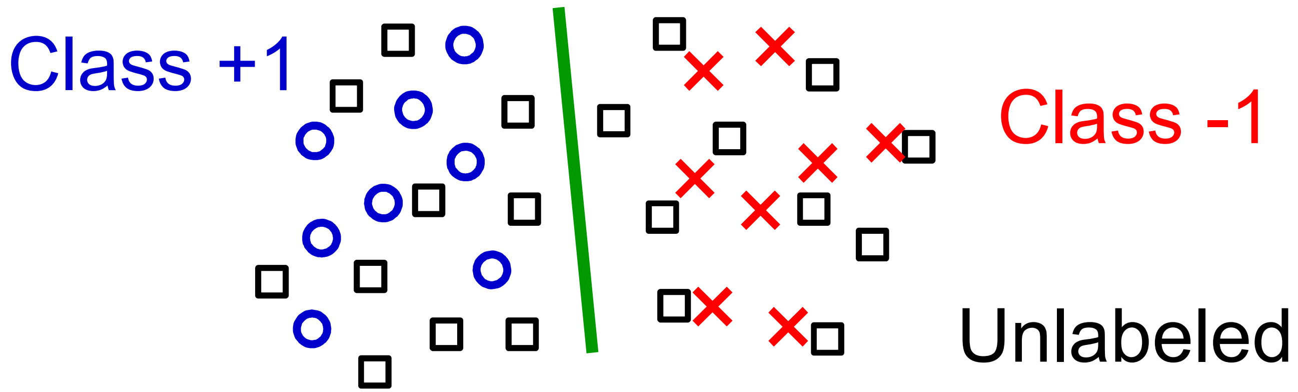


Organization

1. Classification of classification
2. PU classification
3. PNU (=PU+PN) classification
4. UU classification

Semi-Supervised (PNU=PU+PN) Classification

Sakai, du Plessis, Niu & Sugiyama (arXiv2016)



- PU data is enough for optimal learning.
- Convex combination of PU & PN is still optimal!

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

$$R_{\mathrm{PN}}(f) = \pi \int \ell\Big(f(\boldsymbol{x})\Big) p(\boldsymbol{x}|y=+1)\mathrm{d}\boldsymbol{x} + (1-\pi) \int \ell\Big(-f(\boldsymbol{x})\Big) p(\boldsymbol{x}|y=-1)\mathrm{d}\boldsymbol{x}$$

$$R_{\mathrm{PU}}(f) = \pi \int \widetilde{\ell}\Big(f(\boldsymbol{x})\Big) p(\boldsymbol{x}|y=+1)\mathrm{d}\boldsymbol{x} + \int \ell\Big(-f(\boldsymbol{x})\Big) p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \quad \widetilde{\ell}(m) = \ell(m) - \ell(-m)$$

  - Precisely, we switch PU+PN and NU+PN.

PU+PN Classification

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

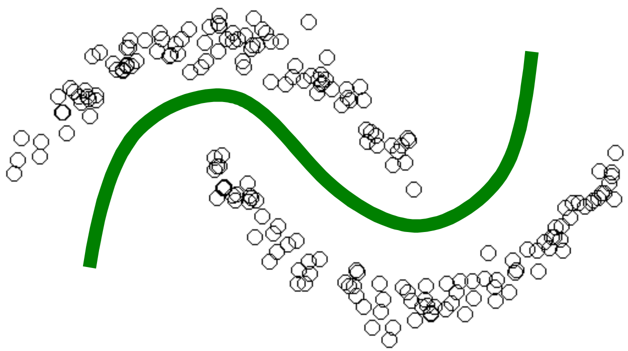
- We use unlabeled data for loss evaluation, not for regularization (as manifold smoothing).
  - Label information is extracted from unlabeled data!
- Generalization error bound:

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

  - Unlabeled data helps without cluster assumptions!

$n_{\mathrm{P}}, n_{\mathrm{N}}, n_{\mathrm{U}}$: # of positive, negative and unlabeled samples

$\widehat{R}^{\gamma}_{\mathrm{PU+PN}}$: Empirical version of $R^{\gamma}_{\mathrm{PU+PN}}$



Numerical Results

- Misclassification error rate: [average (std)]

5% t-test

Dataset	n_u	π	$\widehat{\pi}$	PU+PN	EntReg (Gradvalet & 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

- PU+PN works the best!

Organization

1. Classification of classification
2. PU classification
3. PNU (=PU+PN) classification
4. UU classification

UU Classification: Setup

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

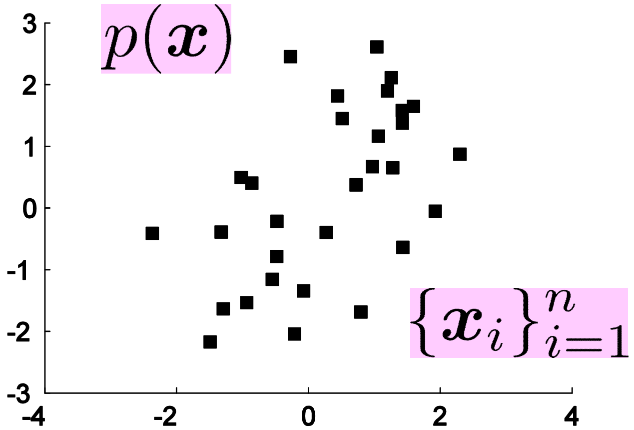
$$\{\boldsymbol{x}_i\}_{i=1}^{n} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}) \qquad \{\boldsymbol{x}'_i\}_{i=1}^{n'} \overset{\text{i.i.d.}}{\sim} p'(\boldsymbol{x})$$

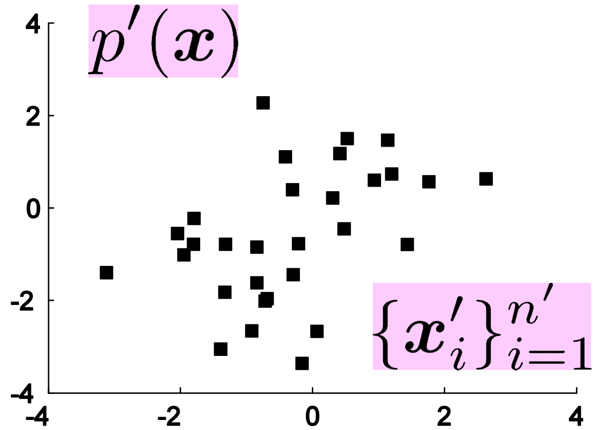
- **Assumption:** Only class-priors are different

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

- **Goal:** Learn a classifier for equal test class-prior

$$q(y = \pm 1) = 1/2$$





Optimal 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 equal test class-prior $q(y=\pm 1) = 1/2$,

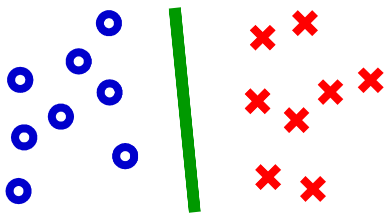
$$g(\boldsymbol{x}) = C\mathrm{sign}[p(\boldsymbol{x}) - p'(\boldsymbol{x})]$$

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

- Sign of C is unknown, but just knowing

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

allows optimal classification!



Decision boundary

Estimation Method 1

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

- Difference of kernel density estimators:
  - Estimate $p(\boldsymbol{x}), p'(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^{n}, \{\boldsymbol{x}'_i\}_{i=1}^{n'}$ separately.
  - Simple but systematic under-estimation of $p(\boldsymbol{x}) - p'(\boldsymbol{x})$.

Anderson, Hall & Titterington (J. Multivariate Analysis 1994)

Estimation Method 2

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

- Direct density-difference estimation:
  - Directly fit a model to $f(\boldsymbol{x}) = p(\boldsymbol{x}) - p'(\boldsymbol{x})$ without explicitly estimating $p(\boldsymbol{x}), p'(\boldsymbol{x})$.
  - Linear least-squares yields an analytic solution:

$$\widehat{f} = \underset{\widetilde{f}}{\mathrm{argmin}} \int \left(\widetilde{f}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 \mathrm{d}\boldsymbol{x}$$

$$= \underset{\widetilde{f}}{\mathrm{argmin}} \int \left(\widetilde{f}(\boldsymbol{x})\right)^2 \mathrm{d}\boldsymbol{x} - 2\int f(\boldsymbol{x})\widetilde{f}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

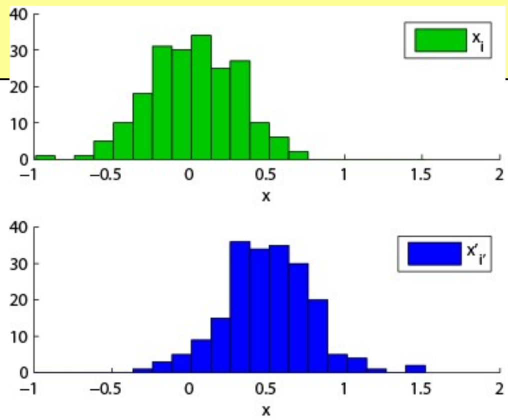
Kim & Scott (IEEE-TPAMI2010)
Sugiyama, Suzuki, Kanamori, du Plessis, Liu & Takeuchi (NIPS2012, NeCo2013)

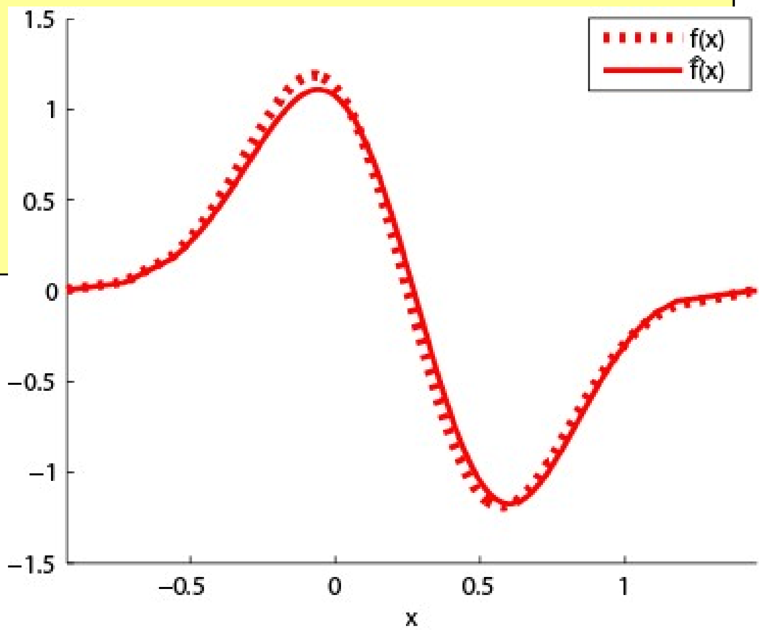
  - $\mathcal{O}\left(n^{-1/2}\right)$ convergence under proper setting.

Least-Squares Density Difference (LSDD): MATLAB® Implementation

- Essentially only 1 line!

```
% Data generation
n=400; x=randn(1,n/2); y=randn(1,n/2)+1; z=[x y];
a=repmat(z.^2,n,1); b=a+a'-2*z'*z; G=sqrt(pi)*exp(-b/4);
h=mean(exp(-b(:,1:n/2)/2),2)-mean(exp(-b(:,n/2+1:n)/2),2);
% Computing the solution
t=(G+0.1*eye(n))\h;
plot(z,G*t,'*');
```





Estimation Method 3

- Direct sign density-difference (DSDD) estimation: du Plessis, Niu & Sugiyama (TAAI2013)
  - $\mathrm{sign}[p(\boldsymbol{x}) - p'(\boldsymbol{x})]$ is the solution of

$$\sup_{r} \int r(\boldsymbol{x})[p(\boldsymbol{x}) - p'(\boldsymbol{x})]\mathrm{d}\boldsymbol{x}$$

$$\text{subject to } |r(\boldsymbol{x})| \leq 1$$

This corresponds to maximizing Fenchel dual lower-bound of L[1]-distance:

$$\int \Big|p(\boldsymbol{x}) - p'(\boldsymbol{x})\Big|\mathrm{d}\boldsymbol{x}$$

Keziou (2003)

  - Empirical version: $\max_{r} \left[\frac{1}{n}\sum_{i=1}^{n} R(\boldsymbol{x}_i) - \frac{1}{n'}\sum_{i'=1}^{n'} R(\boldsymbol{x}'_{i'}) \right]$

$$R(\boldsymbol{x}) = \min\left(1, \max\left(-1, r(\boldsymbol{x})\right)\right)$$

    - Since it is non-convex, we use the convex-concave procedure (CCCP) to obtain a local solution.
    - $\mathcal{O}\left(n^{-1/2}\right)$ convergence under proper setting.

Numerical Results

- Misclassification error rate: [average (std)]

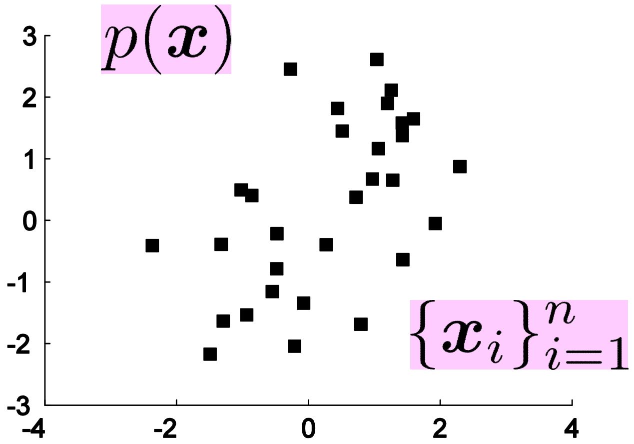
	UU classification			Clustering		
	$\mathrm{sign}[p(\boldsymbol{x})-p'(\boldsymbol{x})]$	$p(\boldsymbol{x})-p'(\boldsymbol{x})$	$p(\boldsymbol{x}),p'(\boldsymbol{x})$	k-means	Spectral Ng et al. (NIPS2001)	Infomax Sugiyama et al. (ICML2011)
Dataset	**DSDD**	**LSDD**	**KDE**	**KM**	**SC**	**SMIC**
australian	**.244** (.116)	.259 (.088)	.355 (.104)	.265 (.080)	.376 (.065)	.308 (.107)
banana	**.338** (.094)	**.339** (.100)	.365 (.067)	.433 (.049)	.427 (.069)	.424 (.070)
diabetes	**.340** (.075)	.361 (.124)	.345 (.034)	.373 (.063)	.380 (.048)	.371 (.114)
german	.375 (.042)	.380 (.093)	**.354** (.057)	.437 (.024)	.445 (.057)	.438 (.041)
heart	.270 (.133)	**.247** (.084)	.354 (.052)	.264 (.059)	.315 (.081)	.327 (.089)
image	**.331** (.078)	.350 (.067)	.350 (.039)	.384 (.031)	.354 (.049)	.382 (.050)
ionosphere	**.291** (.099)	.356 (.066)	.345 (.048)	.330 (.070)	.322 (.058)	.314 (.107)
saheart	.378 (.093)	**.353** (.057)	.363 (.066)	.419 (.082)	.395 (.022)	.385 (.040)
thyroid	**.227** (.098)	.251 (.087)	.302 (.022)	.326 (.061)	.329 (.047)	.307 (.076)
twonorm	.164 (.188)	.153 (.121)	.352 (.096)	**.036** (.053)	.042 (.122)	.049 (.120)

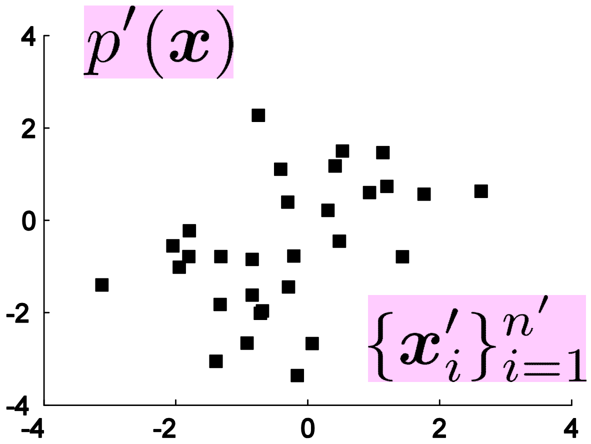
$n = n' = 40$ $\qquad p(y=+1) = 0.35$ $\qquad p'(y=+1) = 0.65$

- UU classification with direct estimation of (sign of) density difference works well !

UU Classification: Summary

du Plessis, Niu & Sugiyama (TAAI2013)





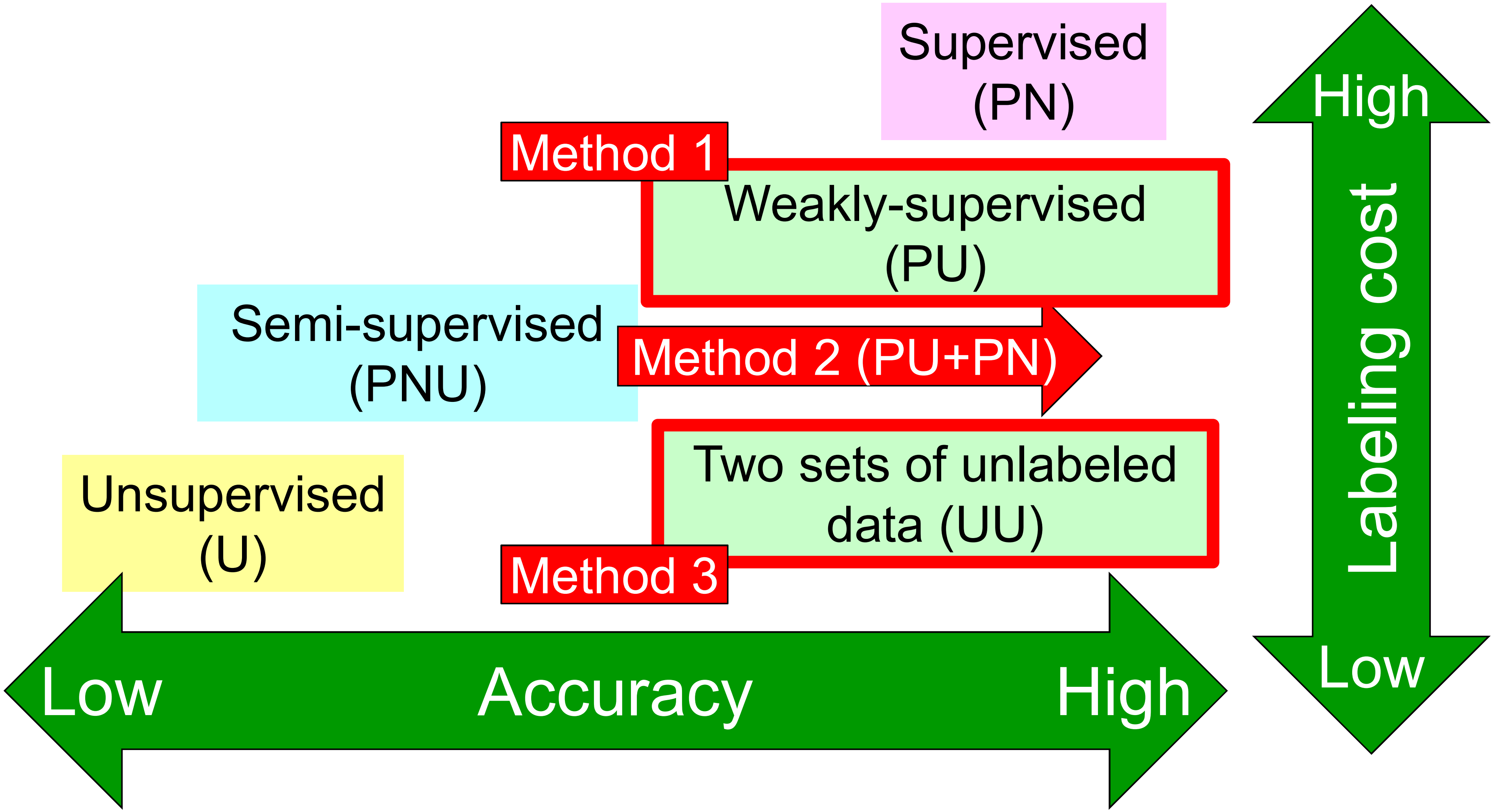
- Given two sets of unlabeled data with different class-priors, estimate the sign of difference of class-posteriors: $\mathrm{sign}[p(\boldsymbol{x}) - p'(\boldsymbol{x})]$
- Same convergence rate as fully supervised case can be achieved!

Organization

1. Classification of classification
2. PU classification
3. PNU (=PU+PN) classification
4. UU classification

Summary

- Classification with high accuracy and low labeling cost is practically important!



RIKEN Center for AIP

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

- RIKEN founded Center for Advanced Intelligence Project (AIP) in 2016.
- Our missions:

1. Development of next-generation AI technology (understand deep learning, go beyond deep learning)
2. Acceleration of scientific research (iPS cells, manufacturing, materials…)
3. Contribution to solving socially critical problems (healthcare for super-aged society, disaster resilience, infrastructure management…)
4. Study of ethical, legal and social issues of AI.
5. Human resource development (academia & industry).

Organization of AIP Center

2017 May

Various application domains
(companies, universities, research institutes, etc.)

Goal-Oriented Technology Research Group:
Abstract complex real-world problems into solvable forms
(22 PIs, 30 researchers, 21 students)

Generic Technology Research Group:
Develop fundamental theory and algorithms
for abstracted problems
(18 PIs, 41 researchers, 30 students)

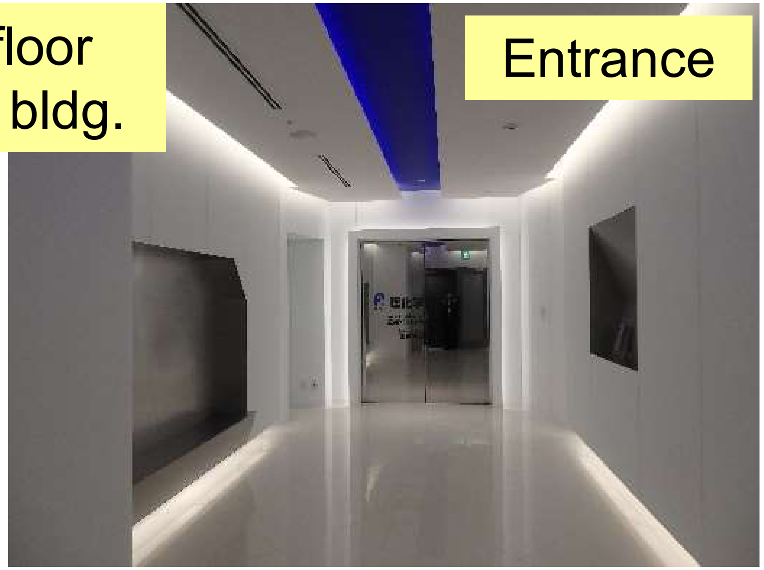
Artificial Intelligence in Society Research Group:
Analyze the influence of AI spreading in society
(8 PIs, 10 researchers)

180
research
staffs
+
17
secre-
taries

Our Office in the Heart of Tokyo!

- Directly connected to Tokyo Metro Nihonbashi Station.
- 6-min walk from Tokyo Station.

Visit us!



Open discussion space