LAMDA
Learning And Mining from DatA
http://lamda.nju.edu.cn

Consistency Theory in Machine Learning

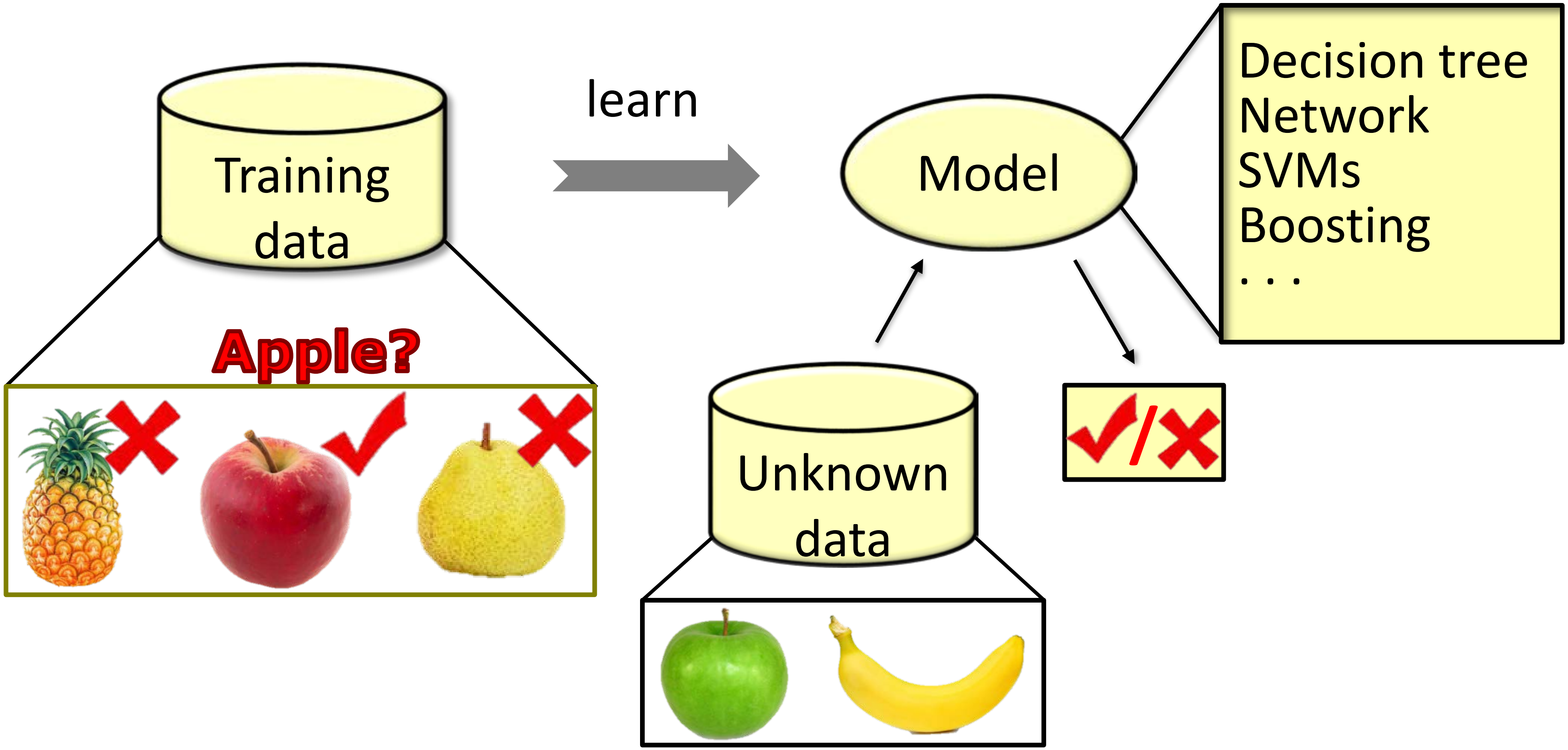
Wei Gao (高 尉)

Learning And Mining from DatA (LAMDA)

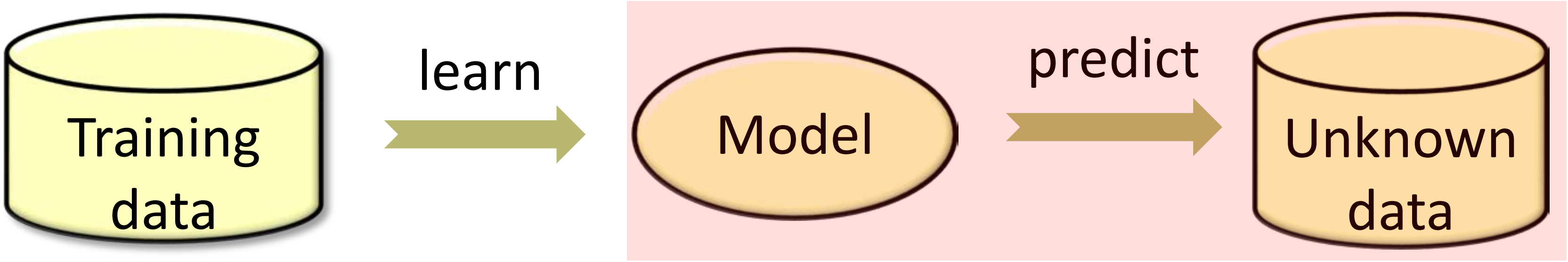
National Key Laboratory for Novel Software Technology

Nanjing University

Machine learning



Generalization



A fundamental problem in machine learning

Generalization: model should predict the unknown data well, not only for the training data

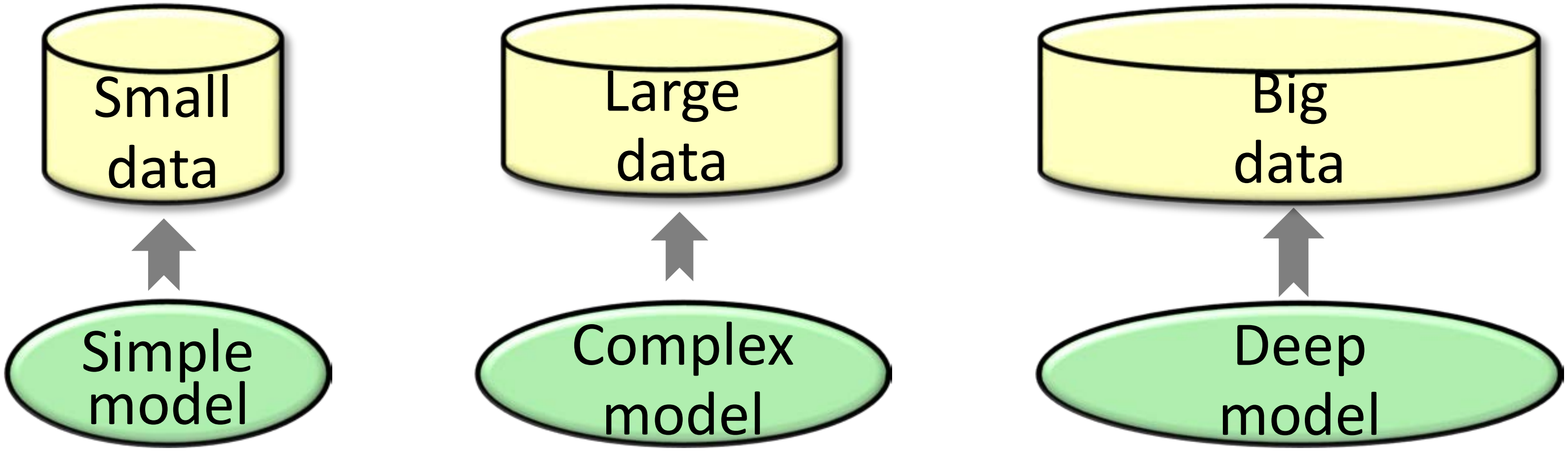
Generalization theoretical analysis

Given model/hypothesis space $\mathcal{H}$, the generalization error of model $h \in \mathcal{H}$ can be bounded by

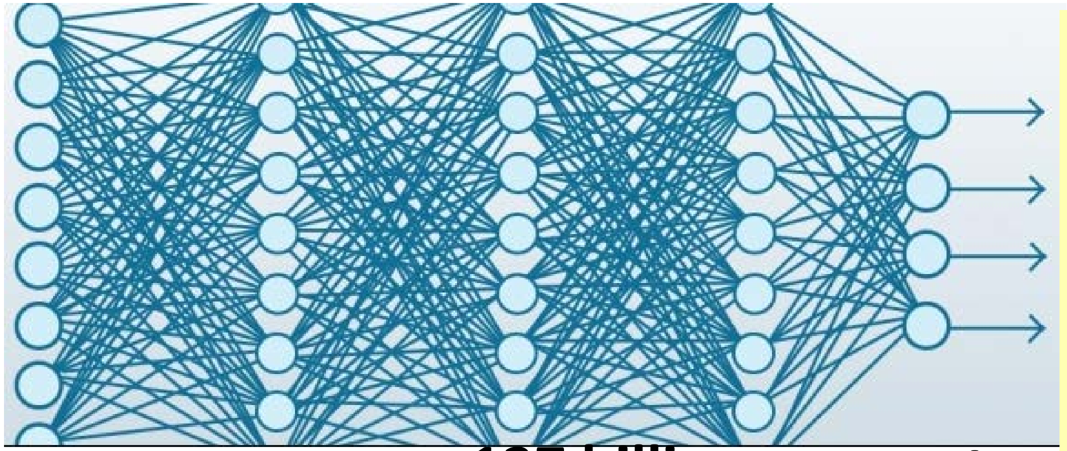
$$\underbrace{\Pr_{\mathcal{D}}[yh(x) < 0]}_{\text{generalization error}} \le \underbrace{\Pr_{S}[yh(x) < 0]}_{\text{empirical error}} + \sqrt{O\left(\frac{\text{model complexity}}{n}\right)}$$

- VC theory [Vapnik & Chervonenkis 1971; Alon et al. 1987; Harvey et al. 2017]
- Cover number [Pollard, 1984; Vapnik, 1998; Golowich et al. 2018]
- Rademacher complexity [Koltchinskii & Panchenko 2000, Bartlett et al. 2017]
- …

Model complexity



Deep neural network [Shazeer et al. 2017]

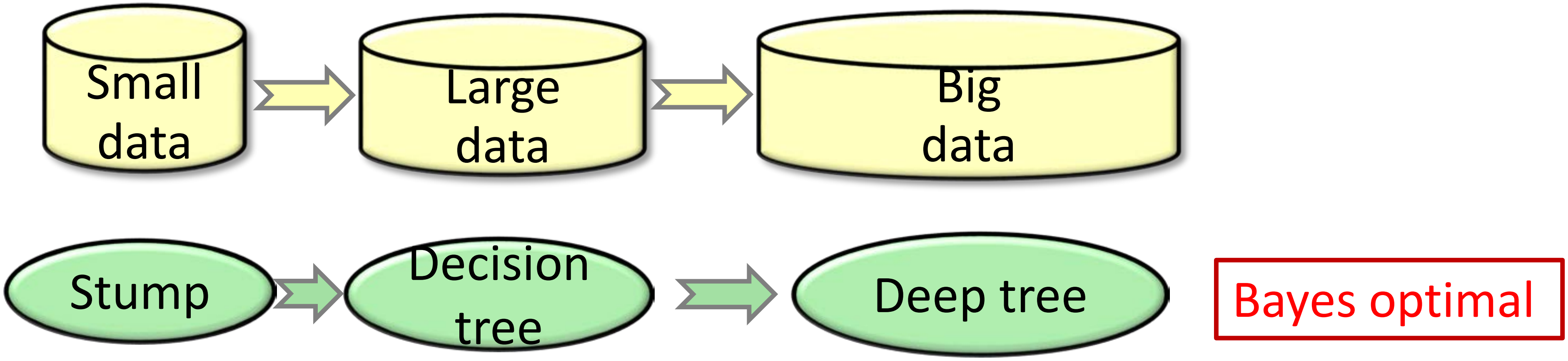


137 billion parameters

Challenges:

- Hard to analyze complexity
- Complexity maybe very high
- Generalization: loose

Consistency



Small data → Large data → Big data

Stump → Decision tree → Deep tree

Bayes optimal

Another important problem in learning theory

Consistency(一致性): model should converge to the Bayes optimal model when training data size $n \to \infty$

- Training data size $n \to \infty$ ⟶ big data
- Model: deep or not deep

Outline

- □ Background on consistency

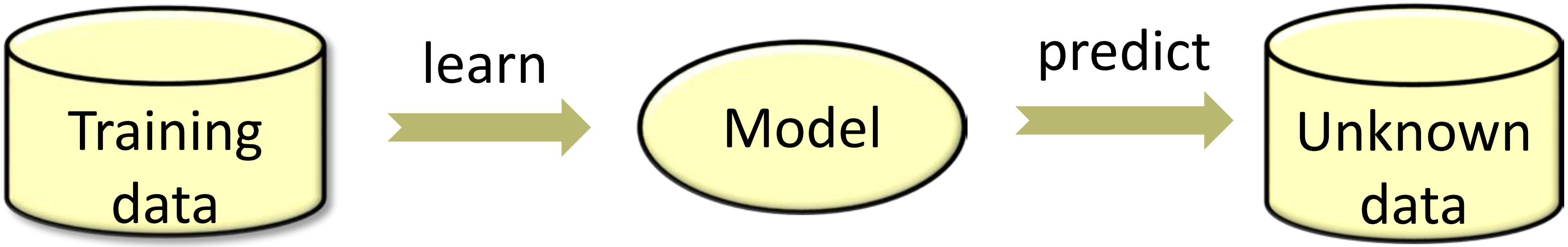
- □ On the consistency of nearest neighbor with noisy data

  Clean data ⟶ Noisy data

- □ On the consistency of pairwise loss

  Univariate loss ⟶ Pairwise loss

Settings



- ➢ Instance space $\mathcal{X}$ and label space $\mathcal{Y}$
- ➢ Unknown distribution D over $\mathcal{X} \times \mathcal{Y}$ unknown data
- ➢ Training data $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ (i.i.d. D)
- ➢ Cost function $c(h(x), y)$ w.r.t. model h and (x, y)

The expected risk of model h is defined as

$$R(h) = E_{(x,y)\sim D}[c(h(x), y)]$$

Bayes risk and consistency

Bayes risk:

$$R^* = \inf_h \{R(h)\} = \inf_h \{E_{(x,y)\sim D}[c(h(x),y)]\}$$

Bayes classifier:

$$h^* = \arg\inf_h \{R(h)\} \quad (R(h^*) = R^*)$$

where the infimum takes over measure functions.

A learning algorithm $\mathcal{A}$ is **consistent** if

$$R(\mathcal{A}_S) \to R^* \quad \text{as training data size } n \to \infty$$

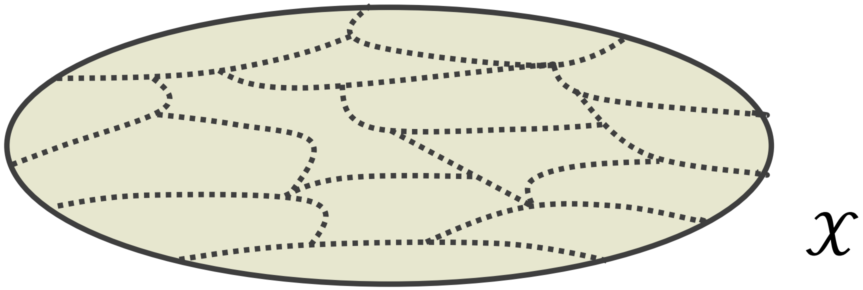
Previous studies on consistency

- Partition algorithms 1951 ~ today
  - Decision tree, k-NN
- Binary classification 1998 ~ today
  - Boosting, SVM…
- Multi-class learning 2004 ~ today
  - Boosting, SVM…
- Multi-label learning 2011 ~ today
  - Boosting, SVM…

Partition algorithms

Partition algorithms

- Partition instance space $\mathcal{X}$ into disjoint cell $A_1, A_2, \ldots, A_n, \ldots$
- Majority vote for each cell



Examples

- Decision tree [Devroye et al. 1997]
- Random forest [Breiman 2000;Biau et al. 2008]
- Nearest neighbor [Fix & Hodges 1951; Cover & Hart 1967]

How about the consistency of partition algorithms?

Consistency on partition algorithms

Stone theorem [Stone 1977]

A partition algorithm is consistent if, as data size $n \to \infty$,

- the diameter of each cell → 0 (in probability)
- the size of train examples in each cell → ∞ (in probability)

k-nearest neighbor is consistent if

$$k = k(n) \to \infty \text{ and } k(n)/n \to 0 \text{ as } n \to \infty$$

Random forest [Biau 2012] is consistent if

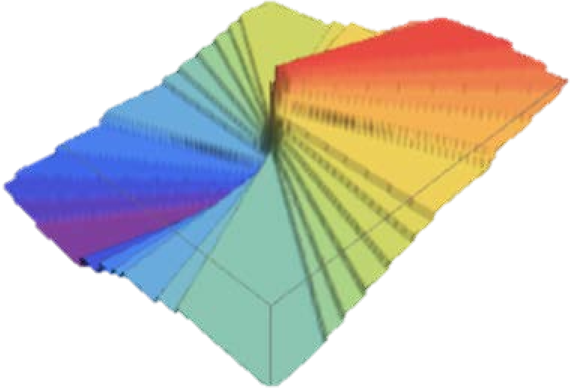
the tree depth $t = t(n) \to \infty$ and $t(n)/n \to 0$ as $n \to \infty$

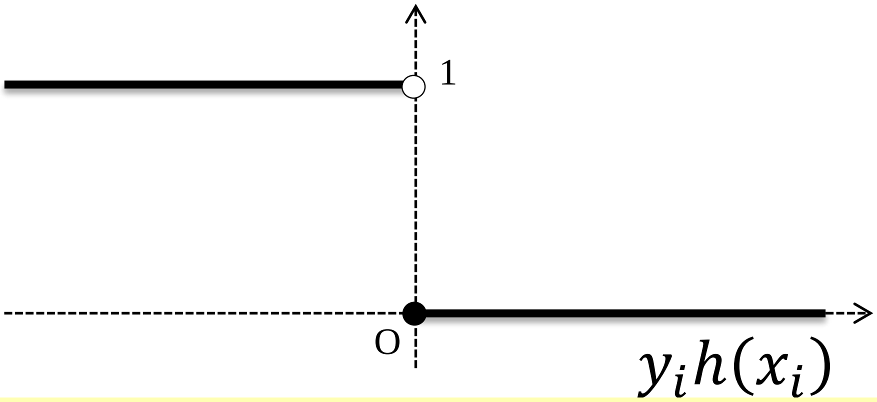
Deep forest is consistent

Binary classification

- Training data $S = \{(x_1, y_1) \ldots (x_n, y_n)\}$
- Real-valued model h: $y = 1$ if $h(x) \geq 0$; otherwise $y = -1$
- The classification error is given by

$$\sum_{i=1}^{n} \frac{I[y_i h(x_i) < 0]}{n}$$



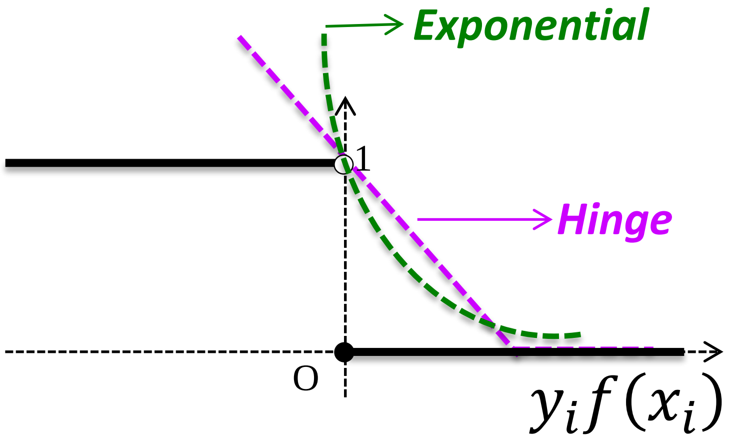


Minimizing such problem is NP-hard (vitally et al. 2012)

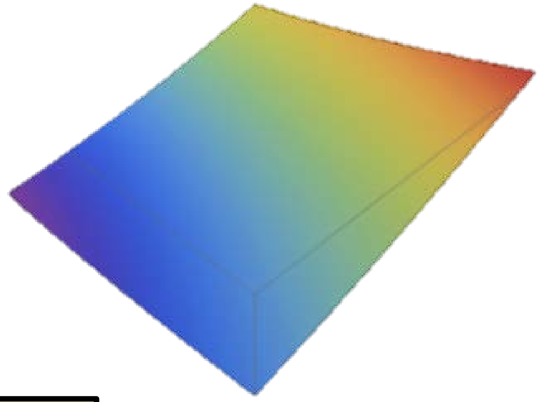
Surrogate loss

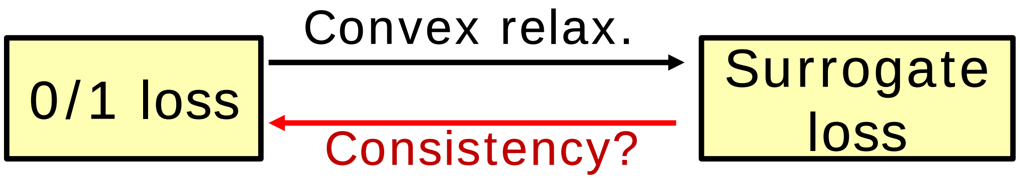
Convex relaxation: ϕ is a convex and continuous surrogate loss

$$\sum_{i=1}^{n}\phi(y_i f(x_i))/n$$



- Boosting: $\phi(t) = e^{-t}$
- SVM: $\phi(t) = \max(0,1-t)$
- Logistic regression: $\phi(t) = \ln(1+e^{-t})$
- …




0/1 loss

Convex relax.

Consistency?

Surrogate loss

Consistency for surrogate loss

A convex surrogate loss ϕ is **calibrated (配准)** if it is differential at 0 with $\phi'(0) < 0$.

Theorem [Bartllet et al. 2007]

The surrogate loss ϕ is **consistent** if and only if it is **calibrated**

- Boosting: $\phi(t) = e^{-t}$
- SVM: $\phi(t) = \max(0,1-t)$
- Least square: $\phi(t) = (1-t)^2$
- Logistic regression: $\phi(t) = \ln(1+e^{-t})$
- …

consistent

Multi-class learning

Label space $\mathcal{Y} = \{1,2,\ldots,L\}$, model $h = (h_1, h_2, \ldots, h_L)$

- **One-vs-one method**: $\sum_i \sum_j \phi\left(h_{y_i}(x_i) - h_j(x_i)\right)$
- **One-vs-all method**: $\sum_i \left(\phi\left(h_{y_i}(x_i)\right) + \sum_{j \neq y_i} \phi\left(-h_j(x_i)\right)\right)$

Consistency for multi-class learning [Zhang 2004; Tewari and Bartlett, 2007]

- Boosting $\phi(t) = e^{-t}$ **consistent**
- Logistic $\phi(t) = \ln(1 + e^{-t})$ **consistent**
- SVM $\phi(t) = \max(0, 1 - t)$ **inconsistent**
- …

Multi-label learning

Multi-label learning predicts a set of labels to an instance

True loss L

- Ranking loss
- Hamming loss
- …

convex relax.

consistency?

Surrogate loss ϕ

- Hinge loss
- Exponential loss
- …

- ➢ Boosting algorithm [Schapire & Singer 2000]
- ➢ Neural network algorithm BP-MIL [Zhang & Zhou 2006]
- ➢ SVM-style algorithms [Elisseeff & Weston 2002; Hariharan et al., 2010]
- ➢ …

How about the consistency for multi-label algorithms?

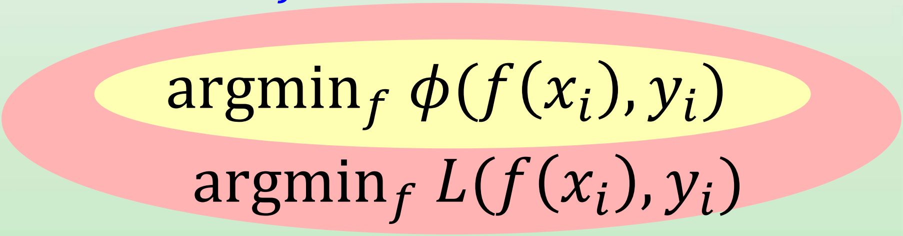
[Gao & Zhou, 2013]

Consistency on multi-label learning

Theorem [Gao & Zhou 2013]

The surrogate loss ϕ is consistent with true loss L if and only if

$$\operatorname{argmin}_f \phi(f(x_i), y_i) \subseteq \operatorname{argmin}_f L(f(x_i), y_i)$$


$\operatorname{argmin}_f \phi(f(x_i), y_i)$
$\operatorname{argmin}_f L(f(x_i), y_i)$

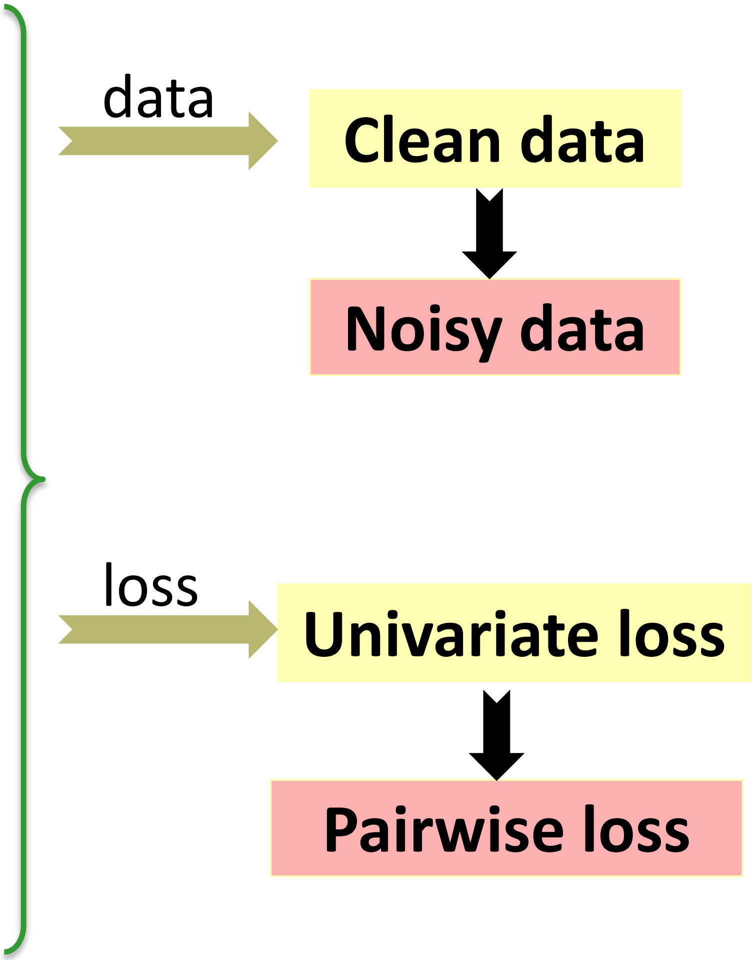
- Boosting algorithm
- Neural network algorithm BP-MIL
- SVM-style algorithms
- …

inconsistent

[Gao & Zhou, 2013]

Previous studies on consistency

- Partition algorithms
  - Decision tree, k-NN
- Binary classification
  - Boosting, SVM…
- Multi-class learning
  - Boosting, SVM…
- Multi-label learning
  - Boosting, NN…



data

Clean data

Noisy data

loss

Univariate loss

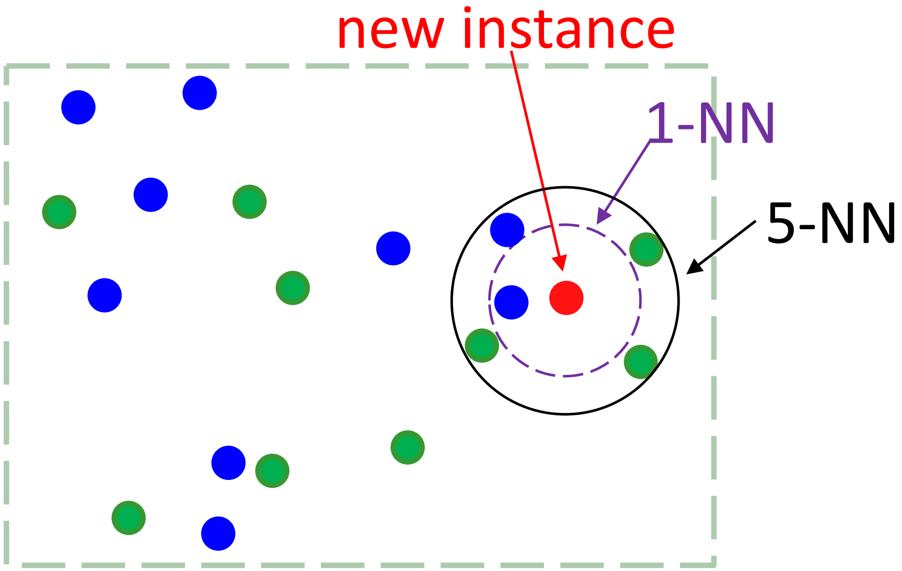
Pairwise loss

Outline

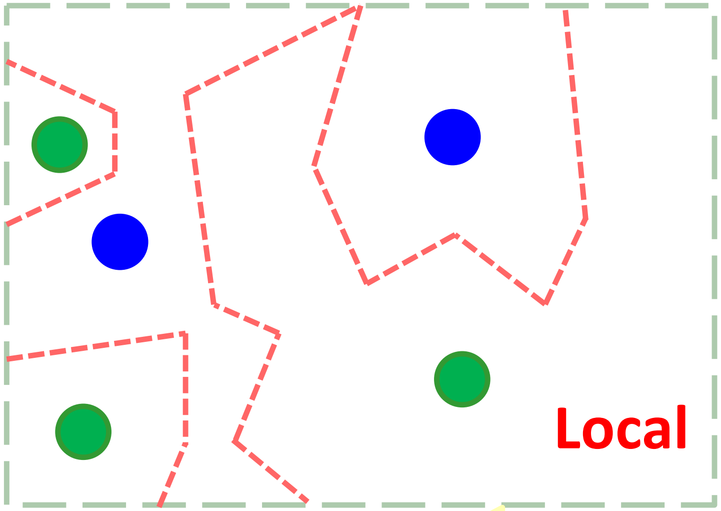
- □ Background on consistency
- □ On the consistency of nearest neighbor with noisy data
- □ On the consistency of pairwise loss

Nearest neighbor (1-NN or k-NN)

Lazy algorithm: classify by the majority vote of k NNs



decision boundary



Local

Consistency on NN [Cover & Hart 1967; Sh

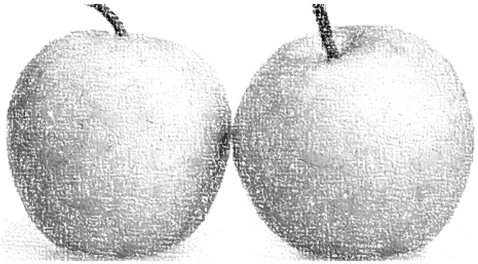
- k-NN (const. k): $+ O(1/\sqrt{k})$
- k-NN ($k = k(n) \to \infty, k/n \to 0$): $R(k(n)$-NN$) \to R^*$

Clean data

Noisy labels

In many real applications:

we collect data whose labels may be corrupted by noise



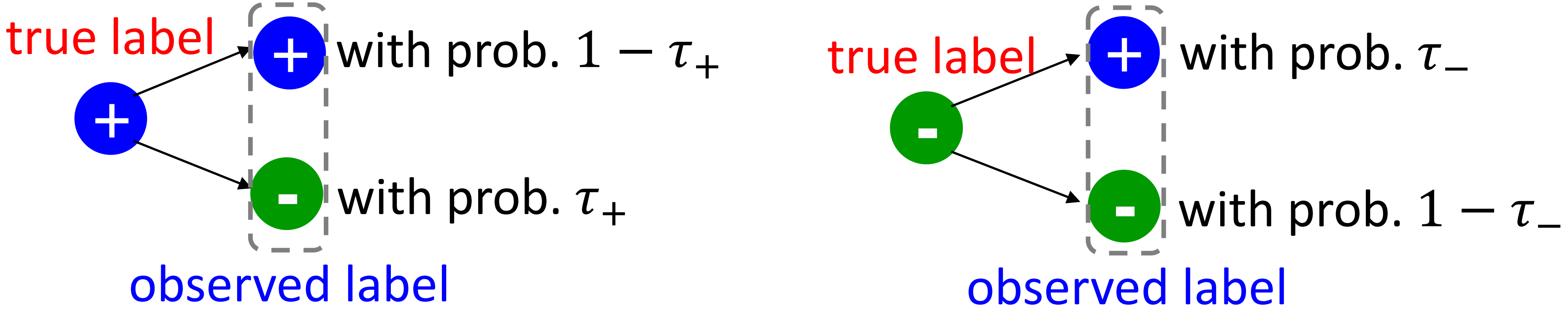
Apple?

Remains open for nearest neighbors with noisy data

Random label noise

Random label noise with rates

$$\tau_+ = \Pr\{\hat{y} = -1 | y = +1\} \text{ and } \tau_- = \Pr\{\hat{y} = +1 | y = -1\}$$



Symmetric noises: $\tau_+ = \tau_-$

Asymmetric noises: $\tau_+ \neq \tau_-$

Consistency of k-NN for symmetric noises

Theorem For symmetric noise with rate $\boldsymbol{\tau}$, let $h_{\hat{S}}^{k}$ be the output of applying k-nearest neighbor to noisy data $\hat{S}$. We have

$$E_{\hat{S}}\left[R\left(\boldsymbol{h}_{\hat{S}}^{k}\right)\right] \leq R^{*} + O\left(\frac{R^{*}}{\sqrt{k}}\right) + O\left(\frac{\boldsymbol{\tau}}{(\mathbf{1}-\mathbf{2}\boldsymbol{\tau})\sqrt{k}}\right) + O\left(\frac{k^{1/(d+1)}}{n^{1/(d+1)}}\right)$$

When $n \to \infty$	Symmetric noise data	Noise-free data
For constant k	$E_{\hat{S}}\left[R\left(h_{\hat{S}}^{k}\right)\right] \to R^{*} + O\left(\frac{1}{\sqrt{k}}\right)$	$E_{\hat{S}}\left[R\left(h_{\hat{S}}^{k}\right)\right] \to R^{*} + O\left(\frac{1}{\sqrt{k}}\right)$
For $k(n) \to \infty$ and $\frac{k}{n} = \frac{k(n)}{n} \to 0$	$E_{\hat{S}}\left[R\left(h_{\hat{S}}^{k}\right)\right] \to R^{*}$	$E_{\hat{S}}\left[R\left(h_{\hat{S}}^{k}\right)\right] \to R^{*}$

k-nearest neighbour is robust to symmetric noise for large k

[Gao et al. ArXiv 2016]

Inconsistency of k-NN for asymmetric noises

Theorem For asymmetric noise with rates τ_+ and τ_-, let $h_{\hat{S}}^k$ be the output of k-nearest neighbor over $\hat{S}$. We have

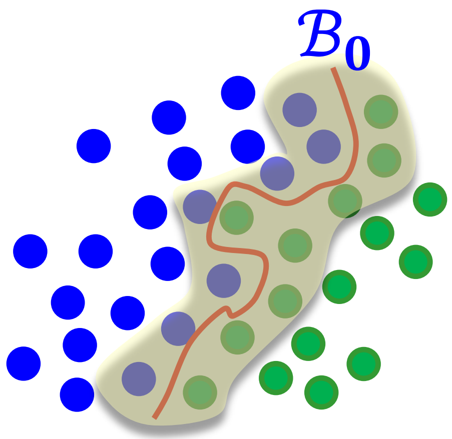
$$E_{\hat{S}}\left[R\left(h_{\hat{S}}^k\right)\right] \to R^* + \Pr[x \in \mathcal{B}_0]$$

for $k = k(n) \to \infty$ and $k/n \to \infty$ as $n \to \infty$

The set of instances whose labels corrupted by asymmetric noise

$$\mathcal{B}_0 = \{x: (\eta(x) - 1/2)(\hat{\eta}(x) - 1/2) < 0\}$$

$\eta(x) = \Pr[y = 1|x]$



Motivation: correct examples in $\mathcal{B}_0$

[Gao et al. ArXiv 2016]

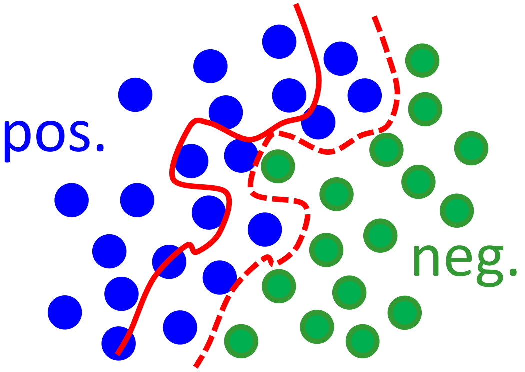
Relation between $\mathcal{B}_0$ and noise rates

Relations between $\eta(x)$ and $\hat{\eta}(x)$:

$$\hat{\eta}(x) - 1/2 = (1 - \tau_+ - \tau_-)(\eta(x) - 1/2) + (\tau_- - \tau_+)/2$$

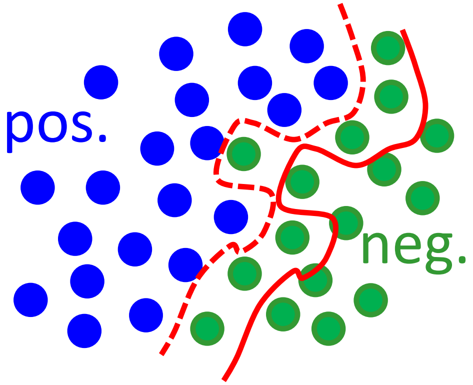
- If $\tau_+ > \tau_-$, then we have

$$\mathcal{B}_0 = \left\{x: \frac{\tau_- - \tau_+}{2} < \hat{\eta}(x) - \frac{1}{2} < 0\right\}$$



- If $\tau_+ < \tau_-$, then we have

$$\mathcal{B}_0 = \left\{x: 0 < \hat{\eta}(x) - \frac{1}{2} < \frac{\tau_- - \tau_+}{2}\right\}$$



How to estimate τ_+ and τ_-?

[Gao et al. ArXiv 2016]

Noise estimation

The noisy conditional probability $\hat{\eta}(x) = \Pr[\hat{y} = 1|x]$

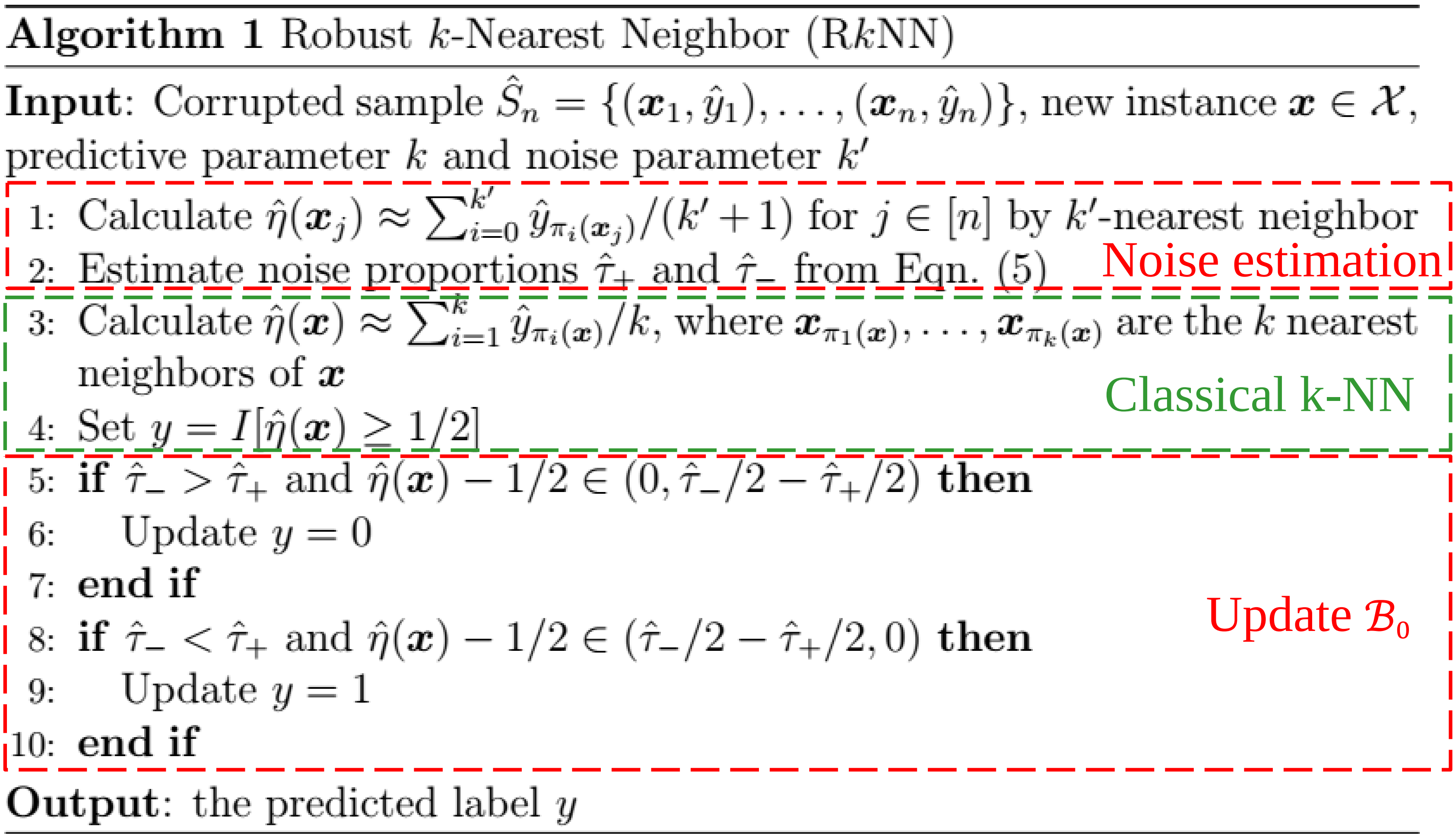
The noise estimation [Liu & Tao 2016; Menon et al., 2015] can be given by

$$\tau_+ = \min_{x \in \hat{S}} \{\hat{\eta}(x)\} \quad \text{and} \quad \tau_- = \min_{x \in \hat{S}} \{1 - \hat{\eta}(x)\}$$

k'-nearest neighbor: estimate $\hat{\eta}(x)$ and calculate τ_+ and τ_-

[Gao et al. ArXiv 2016]

The RkNN algorithm



```
Algorithm 1 Robust $k$-Nearest Neighbor (R$k$NN)
Input: Corrupted sample $\hat{S}_n = \{(\boldsymbol{x}_1, \hat{y}_1), \ldots, (\boldsymbol{x}_n, \hat{y}_n)\}$, new instance $\boldsymbol{x} \in \mathcal{X}$, predictive parameter $k$ and noise parameter $k'$
Calculate $\hat{\eta}(\boldsymbol{x}_j) \approx \sum_{i=0}^{k'} \hat{y}_{\pi_i(\boldsymbol{x}_j)}/(k'+1)$ for $j \in [n]$ by $k'$-nearest neighbor
Estimate noise proportions $\hat{\tau}_+$ and $\hat{\tau}_-$ from Eqn. (5)
Calculate $\hat{\eta}(\boldsymbol{x}) \approx \sum_{i=1}^{k} \hat{y}_{\pi_i(\boldsymbol{x})}/k$, where $\boldsymbol{x}_{\pi_1(\boldsymbol{x})}, \ldots, \boldsymbol{x}_{\pi_k(\boldsymbol{x})}$ are the $k$ nearest neighbors of $\boldsymbol{x}$
Set $y = I[\hat{\eta}(\boldsymbol{x}) \geq 1/2]$
if $\hat{\tau}_- > \hat{\tau}_+$ and $\hat{\eta}(\boldsymbol{x}) - 1/2 \in (0, \hat{\tau}_-/2 - \hat{\tau}_+/2)$ then
    Update $y = 0$
end if
if $\hat{\tau}_- < \hat{\tau}_+$ and $\hat{\eta}(\boldsymbol{x}) - 1/2 \in (\hat{\tau}_-/2 - \hat{\tau}_+/2, 0)$ then
    Update $y = 1$
end if
Output: the predicted label $y$
```

Noise estimation

Classical k-NN

Update $\mathcal{B}_0$

[Gao et al. ArXiv 2016]

Datasets and compared methods

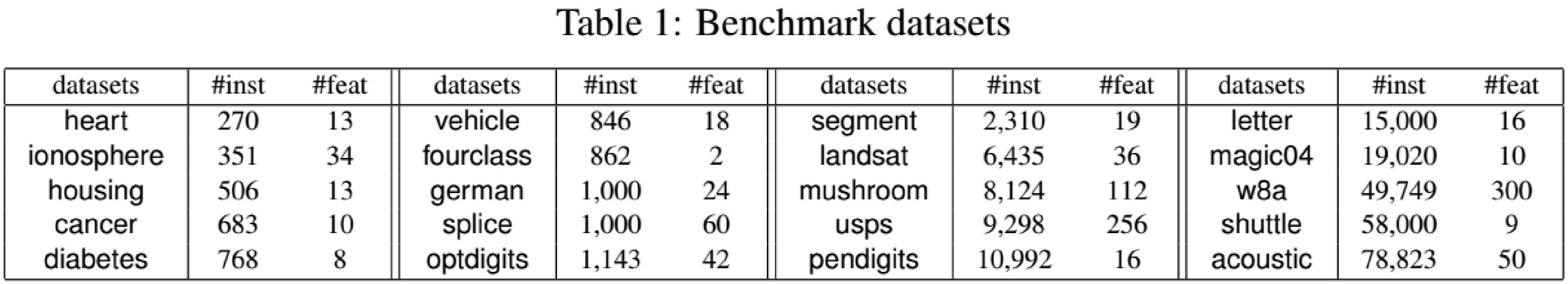


Table 1: Benchmark datasets

datasets	#inst	#feat	datasets	#inst	#feat	datasets	#inst	#feat	datasets	#inst	#feat
heart	270	13	vehicle	846	18	segment	2,310	19	letter	15,000	16
ionosphere	351	34	fourclass	862	2	landsat	6,435	36	magic04	19,020	10
housing	506	13	german	1,000	24	mushroom	8,124	112	w8a	49,749	300
cancer	683	10	splice	1,000	60	usps	9,298	256	shuttle	58,000	9
diabetes	768	8	optdigits	1,143	42	pendigits	10,992	16	acoustic	78,823	50

Compared method

IR-KSVM: kernel Importance-reweighting algorithm [Liu & Tao 2016]

IR-LLog: importance-reweighting algorithm [Liu & Tao 2016]

LD-KSVM: kernel label-dependent algorithm [Natarajan et al. 2013]

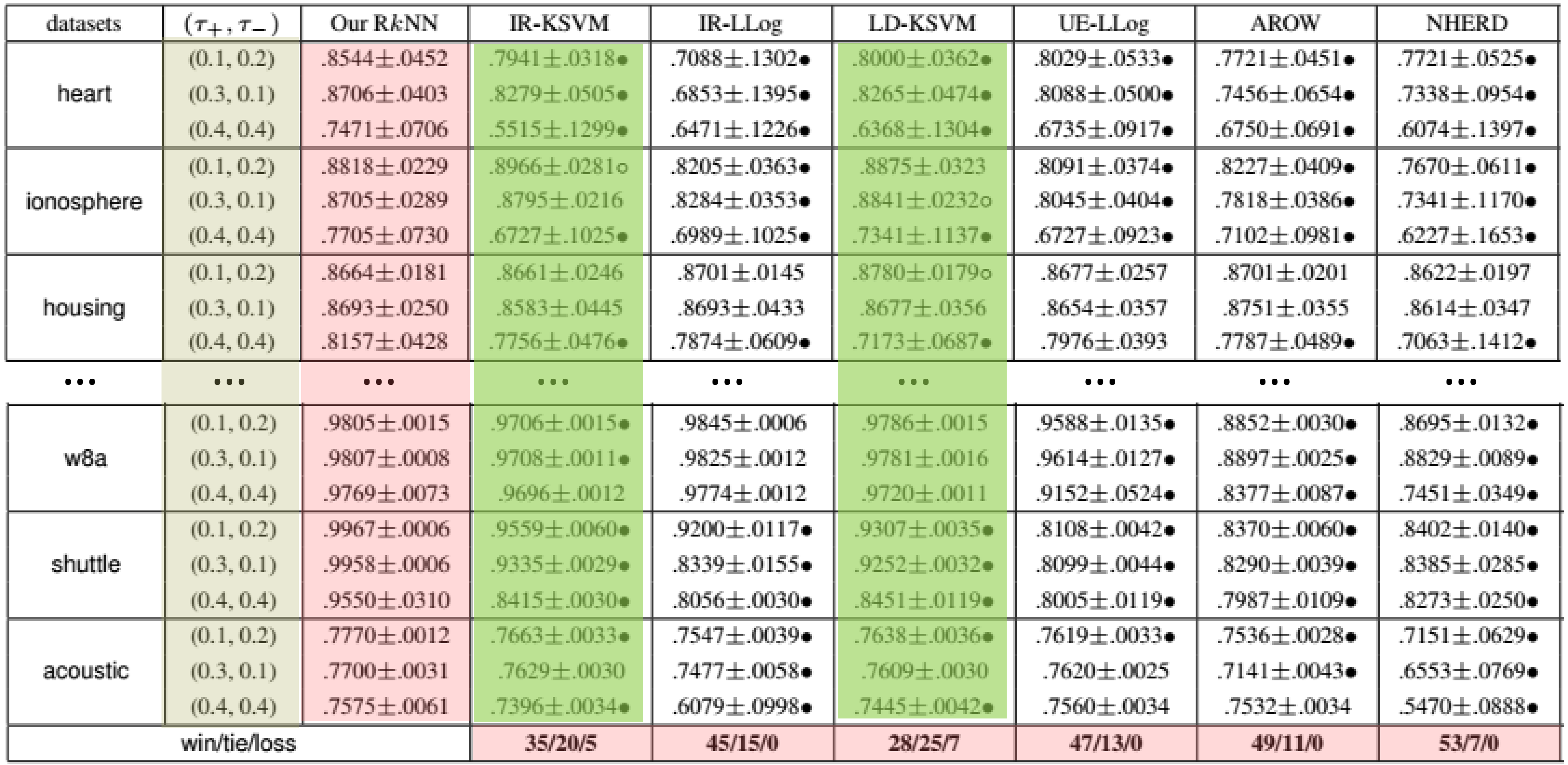
UE-LLog: unbiased-estimator algorithm [Natarajan et al. 2013]

AROW: adaptive regularization of weights [Crammer et al. 2009]

NHERD: normal (Gaussian) herd algorithm [Crammer & Lee 2010]

[Gao et al. ArXiv 2016]

Experimental comparisons



datasets	(τ_+, τ_-)	Our RkNN	IR-KSVM	IR-LLog	LD-KSVM	UE-LLog	AROW	NHERD
heart	(0.1, 0.2)	.8544±.0452	.7941±.0318●	.7088±.1302●	.8000±.0362●	.8029±.0533●	.7721±.0451●	.7721±.0525●
	(0.3, 0.1)	.8706±.0403	.8279±.0505●	.6853±.1395●	.8265±.0474●	.8088±.0500●	.7456±.0654●	.7338±.0954●
	(0.4, 0.4)	.7471±.0706	.5515±.1299●	.6471±.1226●	.6368±.1304●	.6735±.0917●	.6750±.0691●	.6074±.1397●
ionosphere	(0.1, 0.2)	.8818±.0229	.8966±.0281○	.8205±.0363●	.8875±.0323	.8091±.0374●	.8227±.0409●	.7670±.0611●
	(0.3, 0.1)	.8705±.0289	.8795±.0216	.8284±.0353●	.8841±.0232○	.8045±.0404●	.7818±.0386●	.7341±.1170●
	(0.4, 0.4)	.7705±.0730	.6727±.1025●	.6989±.1025●	.7341±.1137●	.6727±.0923●	.7102±.0981●	.6227±.1653●
housing	(0.1, 0.2)	.8664±.0181	.8661±.0246	.8701±.0145	.8780±.0179○	.8677±.0257	.8701±.0201	.8622±.0197
	(0.3, 0.1)	.8693±.0250	.8583±.0445	.8693±.0433	.8677±.0356	.8654±.0357	.8751±.0355	.8614±.0347
	(0.4, 0.4)	.8157±.0428	.7756±.0476●	.7874±.0609●	.7173±.0687●	.7976±.0393	.7787±.0489●	.7063±.1412●
…	…	…	…	…	…	…	…	…
w8a	(0.1, 0.2)	.9805±.0015	.9706±.0015●	.9845±.0006	.9786±.0015	.9588±.0135●	.8852±.0030●	.8695±.0132●
	(0.3, 0.1)	.9807±.0008	.9708±.0011●	.9825±.0012	.9781±.0016	.9614±.0127●	.8897±.0025●	.8829±.0089●
	(0.4, 0.4)	.9769±.0073	.9696±.0012	.9774±.0012	.9720±.0011	.9152±.0524●	.8377±.0087●	.7451±.0349●
shuttle	(0.1, 0.2)	.9967±.0006	.9559±.0060●	.9200±.0117●	.9307±.0035●	.8108±.0042●	.8370±.0060●	.8402±.0140●
	(0.3, 0.1)	.9958±.0006	.9335±.0029●	.8339±.0155●	.9252±.0032●	.8099±.0044●	.8290±.0039●	.8385±.0285●
	(0.4, 0.4)	.9550±.0310	.8415±.0030●	.8056±.0030●	.8451±.0119●	.8005±.0119●	.7987±.0109●	.8273±.0250●
acoustic	(0.1, 0.2)	.7770±.0012	.7663±.0033●	.7547±.0039●	.7638±.0036●	.7619±.0033●	.7536±.0028●	.7151±.0629●
	(0.3, 0.1)	.7700±.0031	.7629±.0030	.7477±.0058●	.7609±.0030	.7620±.0025	.7141±.0043●	.6553±.0769●
	(0.4, 0.4)	.7575±.0061	.7396±.0034●	.6079±.0998●	.7445±.0042●	.7560±.0034	.7532±.0034	.5470±.0888●
win/tie/loss			**35/20/5**	**45/15/0**	**28/25/7**	**47/13/0**	**49/11/0**	**53/7/0**

Our RkNN is comparable to kernel methods

is significantly better than the others

[Gao et al. ArXiv 2016]

Outline

- Background on consistency
- On the consistency of nearest neighbor with noisy data
- On the consistency of pairwise loss

Univariate loss

Most previous consistency studies focus on univariate loss: defined on a single example.

true: $I[yh(x) \leq 0]$ and surrogate: $\phi(yh(x))$

- k-NN, decision tree
- Multi-class learning

- Binary classification
- Multi-label learning

Advantages:

$$E_{(x,y)}\left[I[yh(x) \leq 0]\right] = E_x\left[\boxed{\eta(x)I[h(x) \leq 0] + \left(1-\eta(x)\right)I[h(x) < 0]}\right]$$

$$\Updownarrow$$

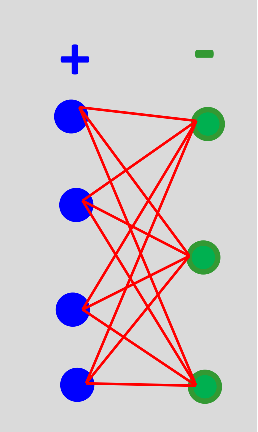
$$E_{(x,y)}\left[\phi\left(yh(x)\right)\right] = E_x\left[\boxed{\eta(x)\phi\left(h(x)\right) + \left(1-\eta(x)\right)\phi\left(-h(x)\right)}\right]$$

Consistency analysis focuses on **single example**

Pairwise loss

In real applications, we aim to optimize the losses, defined on two or multiple examples, such as AUC, F1, Recall, …

AUC: rank positive instances higher than negative instances



Optimizing AUC is over the whole data, rather than single pairwise examples

Challenge:

Consistency analysis for AUC focuses on **the whole data distribution**, rather than singe or two instances

AUC definition

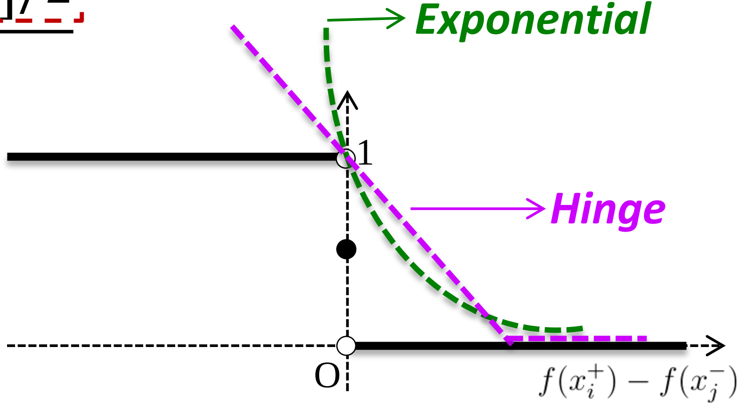
Sample: $S_n = \{(x_1^+, +1) \ldots (x_{n_+}^+, +1), (x_1^-, -1) \ldots (x_{n_-}^-, -1)\}$

The **AUC**, w.r.t. score function h, is defined by

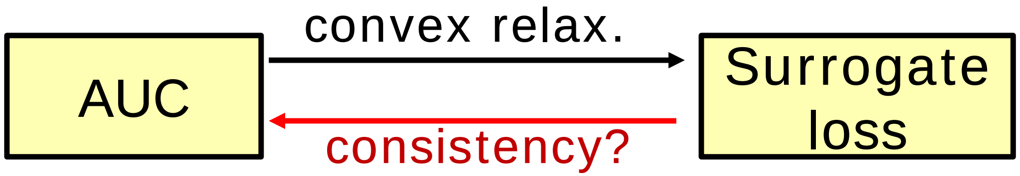
$$\sum_{i=1}^{n_+}\sum_{j=1}^{n_-} \frac{\boxed{I[f(x_i^+) < f(x_j^-)] + I[f(x_i^+) = f(x_j^-)]/2}}{n_+ n_-}$$

$\Big\Downarrow$ surrogate loss

$$\sum_{i=1}^{n_+}\sum_{j=1}^{n_-} \frac{\ell\left(f(x_i^+) - f(x_j^-)\right)}{n_+ n_-}$$



- ✓ Exponential $\ell(t) = e^{-t}$ [Freund et al. 2003; Rudin & Schapire 2009]
- ✓ Hinge $\ell(t) = \max(0, 1-t)$ [Joachims 2006; Zhao et al. 2011]
- ✓ …


AUC

convex relax.

consistency?

Surrogate loss

Least square loss

Least square loss $\ell(t) = (1-t)^2$ is consistent with AUC

Proof sketch: For $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ with margin probability p_i and conditional probability $\xi_i = \Pr[y_i = 1|x_i]$

- Our goal is to minimize the expected risk **over whole distribution**

$$R_\Psi(f) = C_0 + \sum_{i\neq j} p_i p_j \left(\xi_i(1-\xi_j)\ell(f(\mathbf{x}_i) - f(\mathbf{x}_j)) + \xi_j(1-\xi_i)\ell(f(\mathbf{x}_j) - f(\mathbf{x}_i))\right)$$

- Based on sub-gradient conditions, we obtain *n* linear equations

$$\sum_{k\neq i} p_k(\xi_i + \xi_k - 2\xi_i\xi_k)(f(\mathbf{x}_i) - f(\mathbf{x}_k)) = \sum_{k\neq i} p_k(\xi_i - \xi_k) \text{ for each } 1 \leq i \leq n$$

- Solving those linear equations, we get a Bayes solution

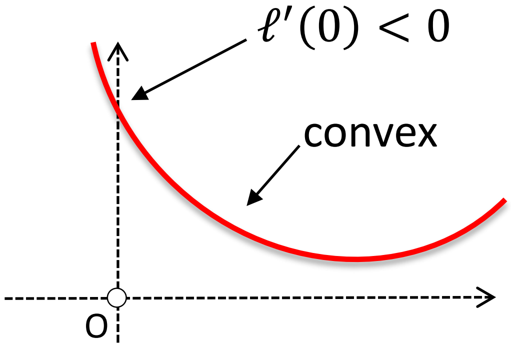
$$f(\mathbf{x}_i) - f(\mathbf{x}_j) = (\xi_i - \xi_j)\frac{\prod_{k\neq i,j}\sum_{l=1}^{n} p_l(\xi_l + \xi_k - 2\xi_l\xi_k)}{\sum_{\substack{s_i\geq 0 \\ s_1+\cdots+s_n=n-2}} p_1^{s_1}\cdots p_n^{s_n}\Gamma(s_1, s_2, \cdots, s_n)}$$

where $\Gamma > 0$ is a polynomial in $(\xi_l + \xi_k - 2\xi_l\xi_k)$

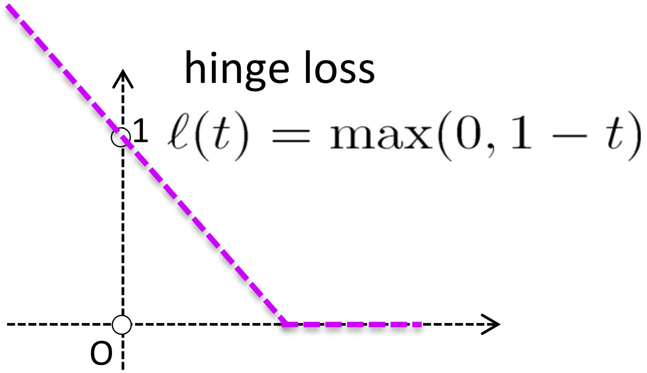
[Gao et al. 2013]

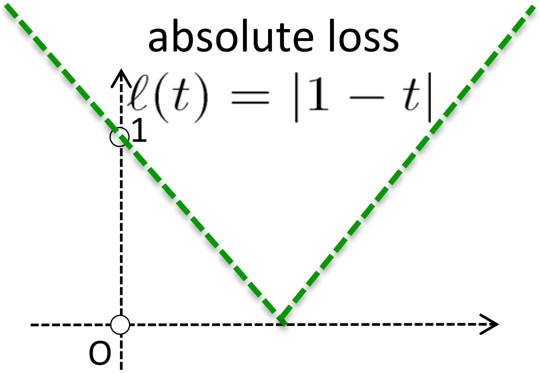
Necessary condition

If a surrogate loss ℓ is consistent with AUC, then loss ℓ is **calibrated** (ℓ is convex with $\ell'(0) < 0$).



Hinge loss and absolute loss are **calibrated** but not consistent with AUC

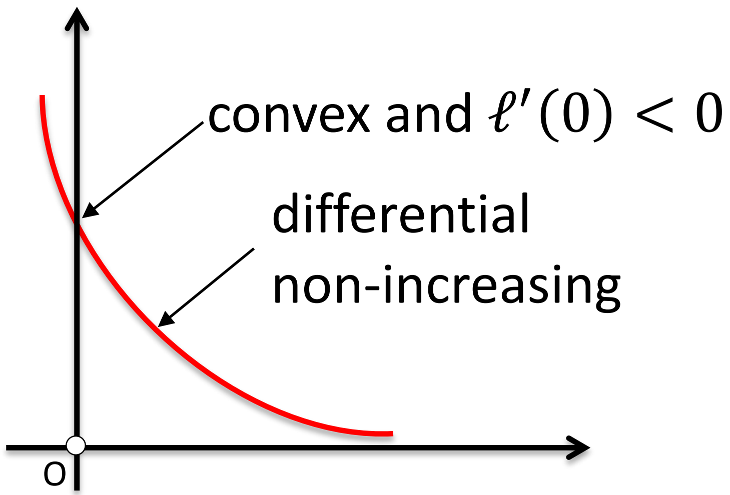




[Gao & Zhou 2015]

Sufficient condition

A surrogate loss ℓ is consistent with AUC if it is calibrated, differential and non-increasing.



Remain open for sufficient and necessary condition

Exponential loss

$$\ell(t) = e^{-t}$$

Logistic loss

$$\ell(t) = \ln(1 + e^{-t})$$

q-norm hinge loss

$$\ell(t) = (\max(0,1 - t))^q$$

Least square hinge loss

$$\ell(t) = (\max(0,1 - t))^2$$

…

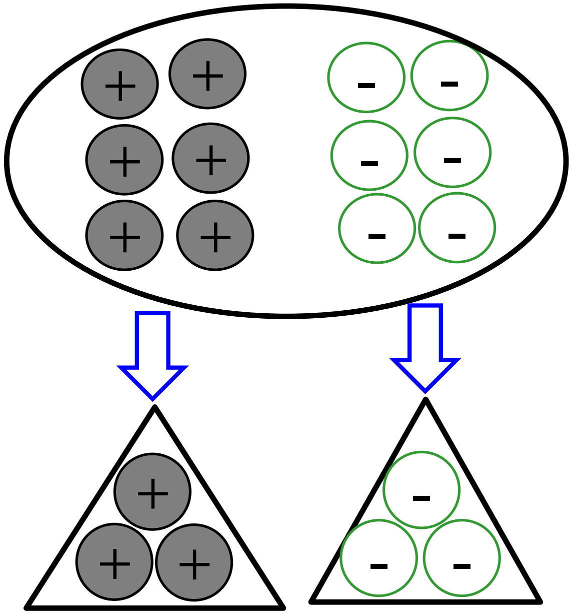
[Gao & Zhou 2015]

Large-scale AUC optimization

Optimize the pairwise loss

$$\sum_{i=1}^{n_+}\sum_{j=1}^{n_-}\ell\left(f(x_i^+)-f\left(x_j^-\right)\right)/n_+n_-$$

- Store all data
- Scan data many time



A simple idea: use a buffer

By using the **hinge loss**, online AUC optimization with a buffer size [Zhao et al., ICML' 2011]

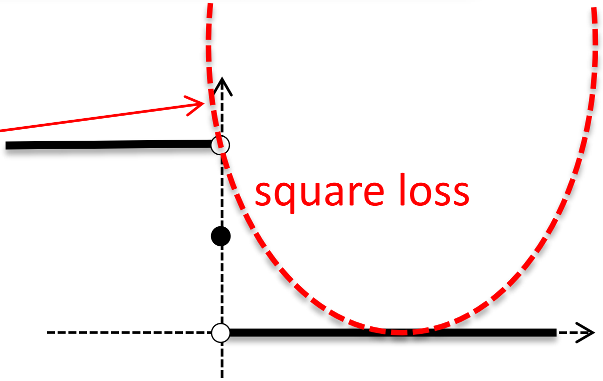
– **hinge loss is inconsistent**

Least square loss

Least square loss $\ell(t) = (1-t)^2$ is consistent with AUC

SGD optimizes

$$\mathcal{L}(w) = \frac{\lambda}{2}|w|^2 + \frac{\sum_{i=1}^{t-1} I[y_i \neq y_t]\boxed{(1-y_t(x_t-x_i)^\top w)^2}}{2|\{i \in [t-1]: y_i y_t = -1\}|}$$



For $y_t = 1$ (similarly for $y_t = -1$)

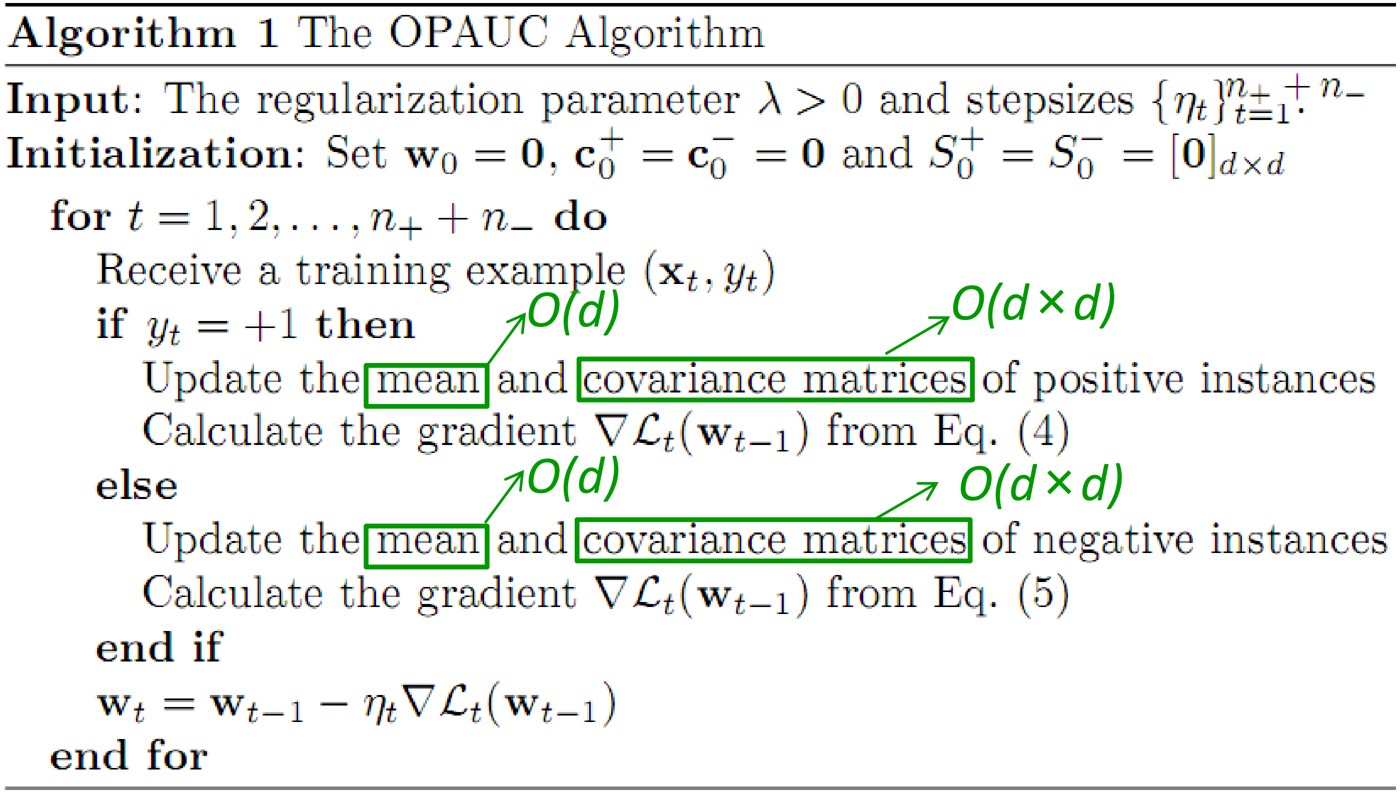
$$\nabla\mathcal{L}(w_{t-1}) = \lambda w - x_t \underbrace{\sum_{i:y_i=-1}\frac{x_i}{n_t^-}}_{\text{neg. mean}} + \left(x_t - \underbrace{\sum_{i:y_i=-1}\frac{x_i}{n_t^-}}_{\text{neg. mean}}\right)\left(x_t - \underbrace{\sum_{i:y_i=-1}\frac{x_i}{n_t^-}}_{\text{neg. mean}}\right)^\top w$$

$$+ \underbrace{\left(\sum_{i:y_i=-1}\frac{x_i x_i^\top}{n_t^-} - \sum_{i:y_i=-1}\frac{x_i}{n_t^-}\sum_{i:y_i=-1}\frac{x_i^\top}{n_t^-}\right)}_{\text{neg. covariance}} w$$

Store the mean and covariance

[Gao et al. 2013, 2016]

OPAUC



Algorithm 1 The OPAUC Algorithm

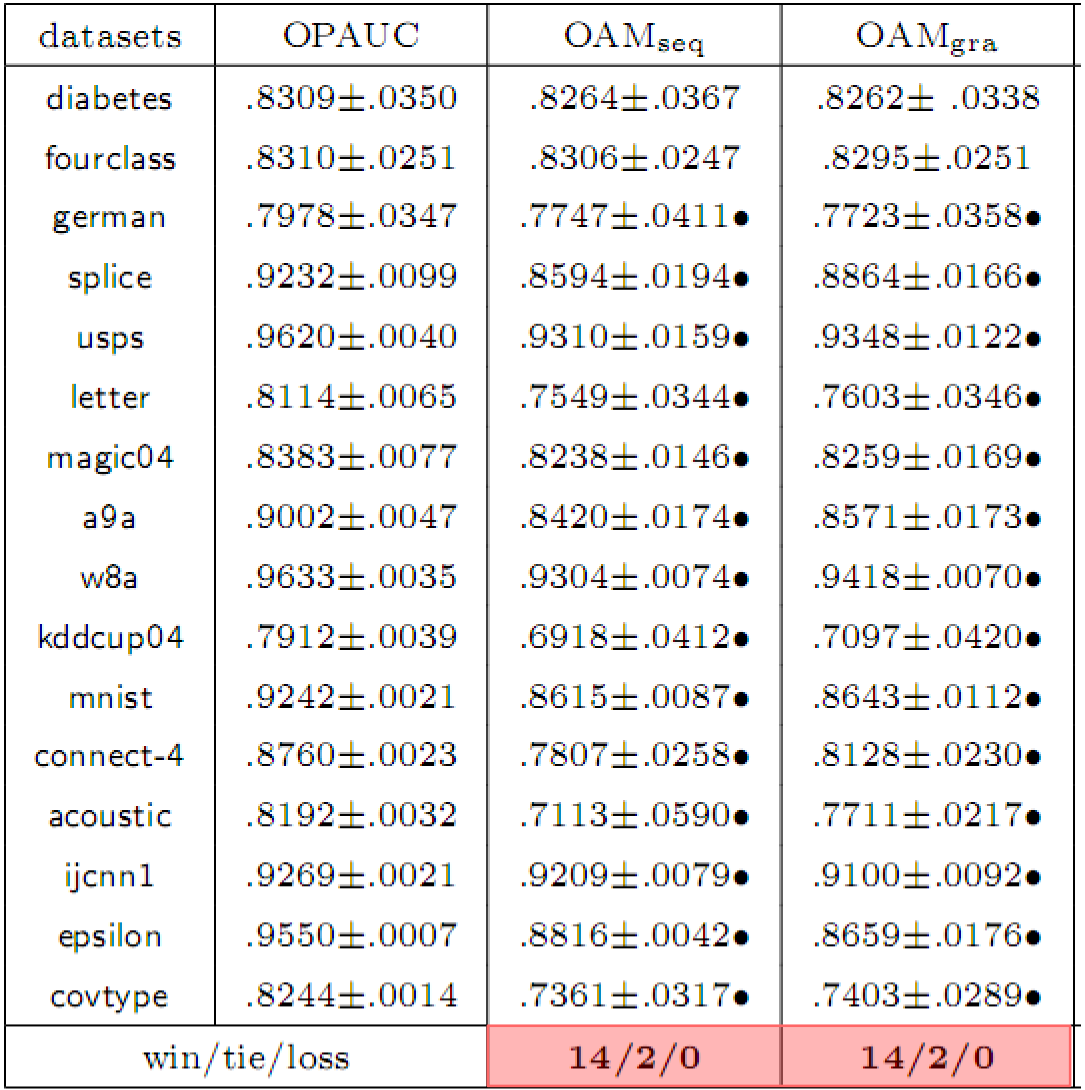
```
Input: The regularization parameter $\lambda > 0$ and stepsizes $\{\eta_t\}_{t=1}^{n_+ + n_-}$.
Initialization: Set $\mathbf{w}_0 = \mathbf{0}$, $\mathbf{c}_0^+ = \mathbf{c}_0^- = \mathbf{0}$ and $S_0^+ = S_0^- = [\mathbf{0}]_{d \times d}$
  for $t = 1, 2, \ldots, n_+ + n_-$ do
    Receive a training example $(\mathbf{x}_t, y_t)$
    if $y_t = +1$ then
      Update the $\overset{O(d)}{\boxed{\text{mean}}}$ and $\overset{O(d \times d)}{\boxed{\text{covariance matrices}}}$ of positive instances
      Calculate the gradient $\nabla \mathcal{L}_t(\mathbf{w}_{t-1})$ from Eq. (4)
    else
      Update the $\overset{O(d)}{\boxed{\text{mean}}}$ and $\overset{O(d \times d)}{\boxed{\text{covariance matrices}}}$ of negative instances
      Calculate the gradient $\nabla \mathcal{L}_t(\mathbf{w}_{t-1})$ from Eq. (5)
    end if
    $\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \nabla \mathcal{L}_t(\mathbf{w}_{t-1})$
  end for
```

Storage: *O(d × d),* independent to data size

Scan data only once

[Gao et al. 2013, 2016]

Results: Existing online methods



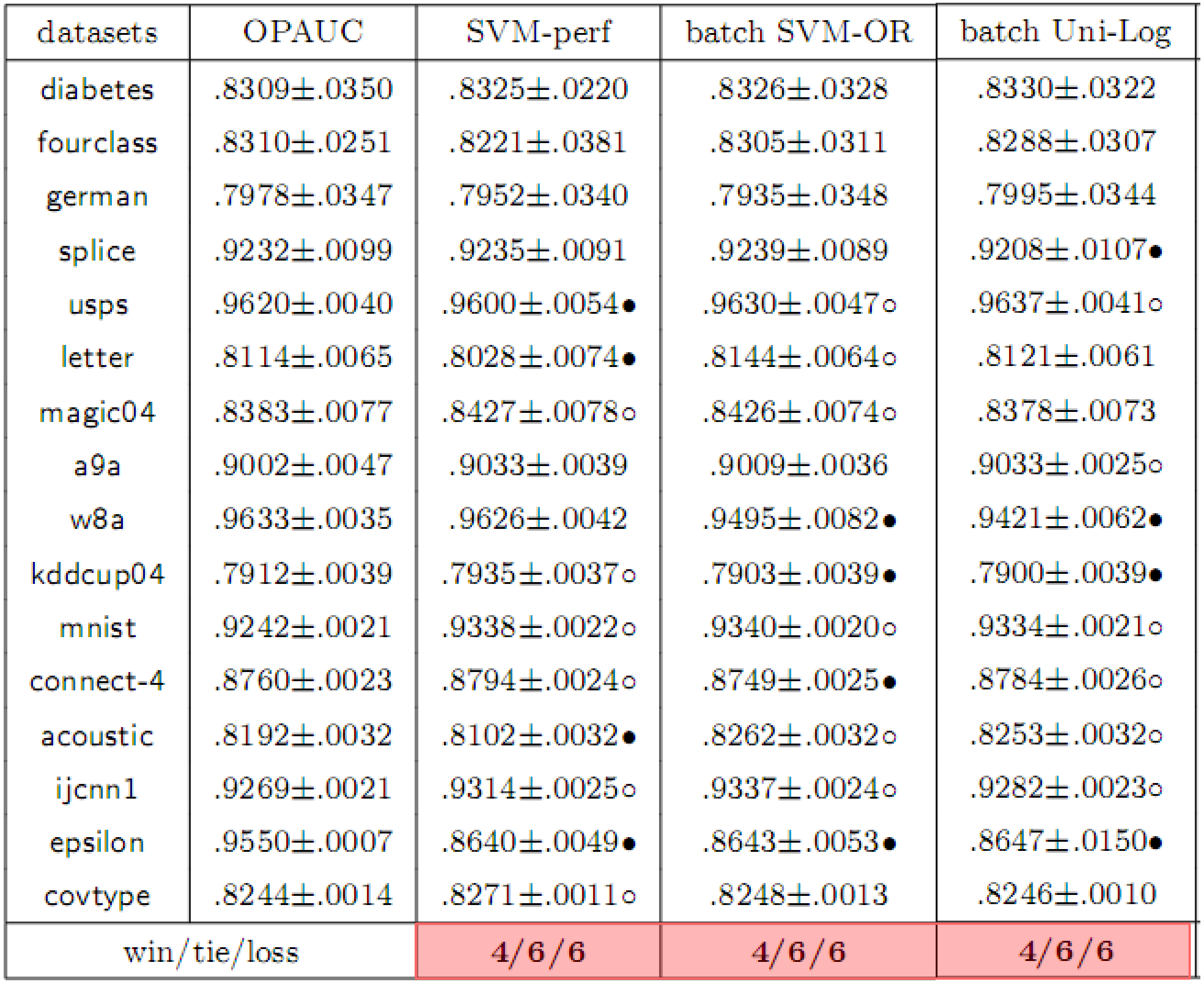
datasets	OPAUC	OAM$_{seq}$	OAM$_{gra}$
diabetes	.8309±.0350	.8264±.0367	.8262± .0338
fourclass	.8310±.0251	.8306±.0247	.8295±.0251
german	.7978±.0347	.7747±.0411●	.7723±.0358●
splice	.9232±.0099	.8594±.0194●	.8864±.0166●
usps	.9620±.0040	.9310±.0159●	.9348±.0122●
letter	.8114±.0065	.7549±.0344●	.7603±.0346●
magic04	.8383±.0077	.8238±.0146●	.8259±.0169●
a9a	.9002±.0047	.8420±.0174●	.8571±.0173●
w8a	.9633±.0035	.9304±.0074●	.9418±.0070●
kddcup04	.7912±.0039	.6918±.0412●	.7097±.0420●
mnist	.9242±.0021	.8615±.0087●	.8643±.0112●
connect-4	.8760±.0023	.7807±.0258●	.8128±.0230●
acoustic	.8192±.0032	.7113±.0590●	.7711±.0217●
ijcnn1	.9269±.0021	.9209±.0079●	.9100±.0092●
epsilon	.9550±.0007	.8816±.0042●	.8659±.0176●
covtype	.8244±.0014	.7361±.0317●	.7403±.0289●
win/tie/loss		**14/2/0**	**14/2/0**

OPAUC significantly better:

- ➢ **Consistency**
- ➢ **buffer**

[Gao et al. 2013, 2016]

Results: Existing batch methods



datasets	OPAUC	SVM-perf	batch SVM-OR	batch Uni-Log
diabetes	.8309±.0350	.8325±.0220	.8326±.0328	.8330±.0322
fourclass	.8310±.0251	.8221±.0381	.8305±.0311	.8288±.0307
german	.7978±.0347	.7952±.0340	.7935±.0348	.7995±.0344
splice	.9232±.0099	.9235±.0091	.9239±.0089	.9208±.0107●
usps	.9620±.0040	.9600±.0054●	.9630±.0047○	.9637±.0041○
letter	.8114±.0065	.8028±.0074●	.8144±.0064○	.8121±.0061
magic04	.8383±.0077	.8427±.0078○	.8426±.0074○	.8378±.0073
a9a	.9002±.0047	.9033±.0039	.9009±.0036	.9033±.0025○
w8a	.9633±.0035	.9626±.0042	.9495±.0082●	.9421±.0062●
kddcup04	.7912±.0039	.7935±.0037○	.7903±.0039●	.7900±.0039●
mnist	.9242±.0021	.9338±.0022○	.9340±.0020○	.9334±.0021○
connect-4	.8760±.0023	.8794±.0024○	.8749±.0025●	.8784±.0026○
acoustic	.8192±.0032	.8102±.0032●	.8262±.0032○	.8253±.0032○
ijcnn1	.9269±.0021	.9314±.0025○	.9337±.0024○	.9282±.0023○
epsilon	.9550±.0007	.8640±.0049●	.8643±.0053●	.8647±.0150●
covtype	.8244±.0014	.8271±.0011○	.8248±.0013	.8246±.0010
win/tie/loss		4/6/6	4/6/6	4/6/6

OPAUC:

- scan once
- store statistics

Batch:

- scan many times
- store whole data

OPAUC highly competitive

[Gao et al. 2013, 2016]

Conclusions

- **Clean data** ⟶ **Noisy data** (k-nearest neighbor)
  - k-NN is consistent for symmetric noise
  - k-NN is biased by asymmetric noise ⟶ RkNN algorithm
- **Univariate loss** ⟶ **Pairwise loss** (AUC)
  - Least square loss is consistent ⟶ OPAUC algorithm
  - Necessary/sufficient condition for AUC consistency

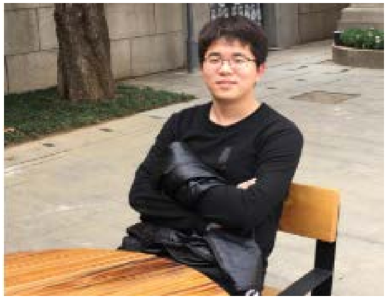
Open problems

- Sufficient and necessary condition for AUC optimization
- Consistency of deep models

感谢

授人以鱼

授人以渔



Thanks for your attention