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

Training data

Model

Decision tree
Network
SVMs
Boosting

Unknown data

Apple?

Learn

http://lamda.nju.edu.cn/gaow/
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

$$\Pr_{\mathcal{D}}[y_h(x) < 0] \leq \Pr_{\mathcal{S}}[y_h(x) < 0] + \sqrt{O\left(\frac{\text{model complexity}}{n}\right)}$$

- **VC theory** [Vapnik & Chervonenkis 1971; Alon et al. 1987; Harvey et al. 2017]
- **Cover number** [Pollard, 1984; Vapnik, 1998; Golowich et al. 2018]
- **...**
Model complexity

- Small data: Simple model
- Large data: Complex model
- Big data: Deep model

Deep neural network [Shazeer et al. 2017]

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

137 billion parameters
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 $n \to \infty \Rightarrow$ big data
- Model: deep or not deep
Outline

- Background on consistency
- On the consistency of nearest neighbor with noisy data
  Clean data $\rightarrow$ Noisy data
- On the consistency of pairwise loss
  Univariate loss $\rightarrow$ Pairwise loss
Settings

- Instance space $\mathcal{X}$ and label space $\mathcal{Y}$
- Unknown distribution $D$ over $\mathcal{X} \times \mathcal{Y}$
- Training data $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ (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

\[ R(\mathcal{A}_n) \rightarrow R^* \quad \text{as training data size } n \rightarrow \infty \]
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 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 $\to 0$ (in probability)
- the size of train examples in each cell $\to \infty$ (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 $t = t(n) \to \infty \text{ and } t(n)/n \to 0 \text{ as } n \to \infty$

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: $\phi$ is a convex and continuous surrogate loss

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

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

Convex relax.  
Consistency?
Consistency for surrogate loss

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

\textbf{Theorem} [Bartlett et al. 2007]

The surrogate loss $\phi$ is \textit{consistent} if and only if it is \textit{calibrated}

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

http://lamda.nju.edu.cn/gaow/
Multi-class learning

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

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

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
- ...

**Surrogate loss $\phi$**
- Hinge loss
- Exponential loss
- ...

Convex relax.

Consistency?

- 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]
Theorem [Gao & Zhou 2013]

The surrogate loss $\phi$ is **consistent** with true loss $L$ if and only if

$$\arg\min_f \phi(f(x_i), y_i) \subseteq \arg\min_f L(f(x_i), y_i)$$
Previous studies on consistency

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

Consistency on NN [Cover & Hart 1967; Shalev-Shwartz & Ben-David 2014]

- $k$-NN (const. $k$): $\mathcal{R}(k)$-NN $\rightarrow \mathcal{R}^*$
- $k$-NN ($k = k(n) \rightarrow \infty$, $k/n \rightarrow 0$): $\mathcal{R}(k(n))$-NN $\rightarrow \mathcal{R}^*$

Clean data
Noisy labels

In many real applications:

we collect data whose **labels** may be corrupted by **noise**

Remains open for nearest neighbors with noisy data
Random label noise

Random label noise with rates

\[ \tau_+ = \Pr\{\hat{y} = -1|y = +1\} \quad \text{and} \quad \tau_- = \Pr\{\hat{y} = +1|y = -1\} \]

Symmetric noises: \( \tau_+ = \tau_- \)

Asymmetric noises: \( \tau_+ \neq \tau_- \)
Consistency of $k$-NN for symmetric noises

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

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

<table>
<thead>
<tr>
<th>When $n \to \infty$</th>
<th>Symmetric noise data</th>
<th>Noise-free data</th>
</tr>
</thead>
<tbody>
<tr>
<td>For constant $k$</td>
<td>$E_{\hat{S}}[R(h_S^k)] \to R^* + O\left(\frac{1}{\sqrt{k}}\right)$</td>
<td>$E_{\hat{S}}[R(h_S^k)] \to R^* + O\left(\frac{1}{\sqrt{k}}\right)$</td>
</tr>
<tr>
<td>For $k(n) \to \infty$ and $k(n)/n = k(n)/n \to 0$</td>
<td>$E_{\hat{S}}[R(h_S^k)] \to R^*$</td>
<td>$E_{\hat{S}}[R(h_S^k)] \to R^*$</td>
</tr>
</tbody>
</table>

$k$-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 $\tau_+$ and $\tau_-$, let $h_{\hat{S}}^k$ be the output of $k$-nearest neighbor over $\hat{S}$. We have

$$E_{\hat{S}}[R(h_{\hat{S}}^k)] \to R^* + \Pr[x \in B_0]$$

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

$$B_0 = \{x : (\eta(x) - 1/2)(\hat{\eta}(x) - 1/2) < 0\}$$

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

Motivation: correct examples in $B_0$

[Gao et al. ArXiv 2016]
Relation between $B_0$ 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
  $$B_0 = \left\{ x : \frac{\tau_- - \tau_+}{2} < \hat{\eta}(x) - \frac{1}{2} < 0 \right\}$$

- If $\tau_+ < \tau_-$, then we have
  $$B_0 = \left\{ x : 0 < \hat{\eta}(x) - \frac{1}{2} < \frac{\tau_- - \tau_+}{2} \right\}$$

How to estimate $\tau_+$ and $\tau_-$?

[Gao et al. ArXiv 2016]
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 \( \tau_+ \) and \( \tau_- \)
The RkNN algorithm

Algorithm 1 Robust k-Nearest Neighbor (RkNN)

**Input:** Corrupted sample $\hat{S}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, new instance $x \in \mathcal{X}$, predictive parameter $k$ and noise parameter $k'$

1. Calculate $\hat{\eta}(x_j) \approx \sum_{i=0}^{k'} \hat{y}_{\pi_i}(x_j) / (k' + 1)$ for $j \in [n]$ by $k'$-nearest neighbor
2. Estimate noise proportions $\hat{\tau}_+$ and $\hat{\tau}_-$ from Eqn. (5)
3. Calculate $\hat{\eta}(x) \approx \sum_{i=1}^{k} \hat{y}_{\pi_i}(x) / k$, where $x_{\pi_1(x)}, \ldots, x_{\pi_k(x)}$ are the $k$ nearest neighbors of $x$
4. Set $y = I[\hat{\eta}(x) \geq 1/2]$
5. if $\hat{\tau}_- > \hat{\tau}_+$ and $\hat{\eta}(x) - 1/2 \in (0, \hat{\tau}_-/2 - \hat{\tau}_+/2)$ then
6. Update $y = 0$
7. end if
8. if $\hat{\tau}_- < \hat{\tau}_+$ and $\hat{\eta}(x) - 1/2 \in (\hat{\tau}_-/2 - \hat{\tau}_+/2, 0)$ then
9. Update $y = 1$
10. end if

**Output:** the predicted label $y$

Datasets and compared methods

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

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

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

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:

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

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

Challenge:
Consistency analysis for AUC focuses on the whole data distribution, rather than single 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

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

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

✓ **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** \( \rightarrow \) **surrogate loss**

**Exponential** \( \rightarrow \) **Hinge**
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

$R_\Psi(f) = C_0 + \sum_{i \neq j} p_i p_j (\xi_i (1 - \xi_j) \ell(f(x_i) - f(x_j)) + \xi_j (1 - \xi_i) \ell(f(x_j) - f(x_i)))$

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

$\sum_{k \neq i} p_k (\xi_i + \xi_k - 2\xi_i \xi_k) (f(x_i) - f(x_k)) = \sum_{k \neq i} p_k (\xi_i - \xi_k)$ for each $1 \leq i \leq n$

- Solving those linear equations, we get a Bayes solution

$f(x_i) - f(x_j) = (\xi_i - \xi_j) \frac{\prod_{k \neq i,j} \sum_{s_{i} \geq 0} p_l (\xi_i + \xi_k - 2\xi_i \xi_k)}{\sum_{s_1 + \cdots + s_n = n-2} p_1^{s_1} \cdots p_n^{s_n} \Gamma(s_1, s_2, \cdots, s_n)}$

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

[Gao et al. 2013]
Necessary condition

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

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

[http://lamda.nju.edu.cn/gaow/]

[Gao & Zhou 2015]
A surrogate loss $\ell$ is consistent with AUC if it is calibrated, differential and non-increasing.

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

...
Large-scale AUC optimization

Optimize the pairwise loss

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

- 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

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

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

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

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

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 $w_0 = 0$, $c_0^+ = c_0^- = 0$ and $S_0^+ = S_0^- = [0]_{d \times d}$

for $t = 1, 2, \ldots, n_+ + n_-$ do

Receive a training example $(x_t, y_t)$

if $y_t = +1$ then

Update the mean and covariance matrices of positive instances

Calculate the gradient $\nabla \mathcal{L}_t(w_{t-1})$ from Eq. (4)

else

Update the mean and covariance matrices of negative instances

Calculate the gradient $\nabla \mathcal{L}_t(w_{t-1})$ from Eq. (5)

end if

$w_t = w_{t-1} - \eta_t \nabla \mathcal{L}_t(w_{t-1})$

end for

Storage: $O(d \times d)$, independent to data size

Scan data only once

[Gao et al. 2013, 2016]
## Results: Existing online methods

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

**OPAUC significantly better:**
- **Consistency**
- **buffer**

[Gao et al. 2013, 2016]
Results: Existing batch methods

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

| 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 → \(Rk\)NN 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

http://lamda.nju.edu.cn/gaow/
Thanks for your attention