

Machine Learning (機器學習)

Lecture 12: Bagging and Boosting

Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

Department of Computer Science
& Information Engineering

National Taiwan University
(國立台灣大學資訊工程系)



Roadmap

- ① When Can Machines Learn?
- ② Why Can Machines Learn?
- ③ How Can Machines Learn?
- ④ How Can Machines Learn Better?
- ⑤ Embedding Numerous Features: Kernel Models
- ⑥ Combining Predictive Features: Aggregation Models

Lecture 12: Bagging and Boosting

- Uniform Blending
- Linear and Any Blending
- Bagging (Bootstrap Aggregation)
- Motivation of Boosting
- Diversity by Re-weighting
- Adaptive Boosting Algorithm
- Adaptive Boosting in Action

An Aggregation Story

Your T friends g_1, \dots, g_T predicts whether stock will go up as $\underline{g_t(\mathbf{x})}$.

You can ...

- **select** the most trust-worthy friend from their **usual performance**
—validation!
- **mix** the predictions from all your friends **uniformly**
—let them **vote!** 投票
- **mix** the predictions from all your friends **non-uniformly**
—let them vote, but **give some more ballots** weight
- **combine** the predictions **conditionally**
—if [t satisfies some condition] give some ballots to friend t
- ...

aggregation models: **mix** or **combine**
hypotheses (for better performance)

Aggregation with Math Notations

Your T friends g_1, \dots, g_T predicts whether stock will go up as $g_t(\mathbf{x})$.

做選擇

- **select** the most trust-worthy friend from their **usual performance**

$$G(\mathbf{x}) = g_{t_*}(\mathbf{x}) \text{ with } t_* = \operatorname{argmin}_{t \in \{1, 2, \dots, T\}} \operatorname{Eval}(g_t^-) \leftarrow \text{選 Eval 最小}$$

- **mix** the predictions from all your friends **uniformly** *同人的結果*

$$G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^T \boxed{1} \cdot g_t(\mathbf{x})\right) \quad \text{公平投票}$$

- **mix** the predictions from all your friends **non-uniformly**

$$G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^T \alpha_t \cdot g_t(\mathbf{x})\right) \text{ with } \alpha_t \geq 0 \quad \text{不公平投票}$$

- {
- include **select**: $\alpha_t = [\underline{\operatorname{Eval}(g_t^-)} \text{ smallest}]$ *給這個人幾張票*
 - include **uniformly**: $\alpha_t = \boxed{1} \leftarrow \text{- 人 - 票}$

- **combine** the predictions **conditionally**

$$G(\mathbf{x}) = \operatorname{sign}\left(\sum_{t=1}^T q_t(\mathbf{x}) \cdot g_t(\mathbf{x})\right) \text{ with } \boxed{q_t(\mathbf{x}) \geq 0} \quad \begin{array}{l} \text{某種 condition} \\ \text{下, 才加投票} \end{array}$$

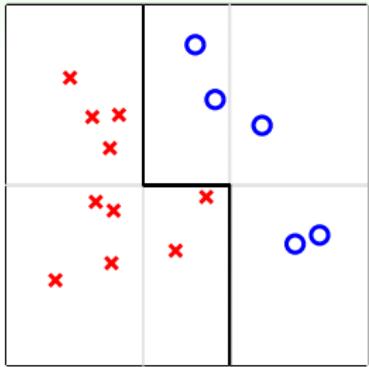
- include **non-uniformly**: $q_t(\mathbf{x}) = \alpha_t$

aggregation models: a **rich family**

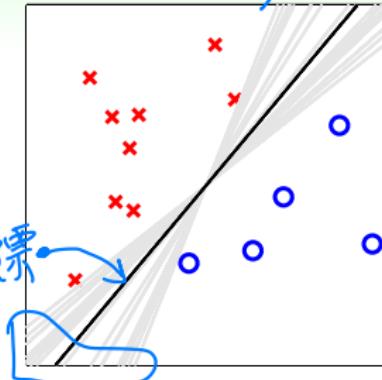
Why Might Aggregation Work?

if 只能用 vertical and horizontal

無限多條 decision boundary



這些線的投票
結果平均



- mix different weak hypotheses uniformly
— $G(\mathbf{x})$ ‘strong’
- aggregation
 \Rightarrow feature transform (?)

- mix different random-PLA hypotheses uniformly
— $G(\mathbf{x})$ ‘moderate’
- aggregation
 \Rightarrow regularization (?)

proper aggregation \Rightarrow better performance

Uniform Blending (Voting) for Classification

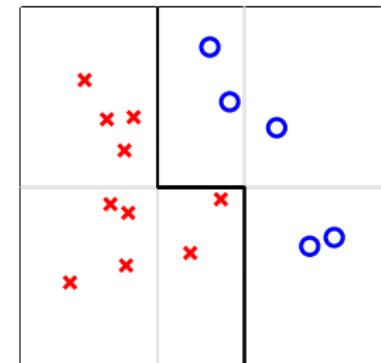
uniform blending: known g_t each with 1 ballot

$$G(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^T 1 \cdot g_t(\mathbf{x}) \right) - \text{票}$$

- same g_t (autocracy):
as good as one single g_t
- very different g_t (diversity + democracy):
majority can correct minority
- similar results with uniform voting for
multiclass

$$G(\mathbf{x}) = \operatorname{argmax}_{1 \leq k \leq K} \sum_{t=1}^T \llbracket g_t(\mathbf{x}) = k \rrbracket$$

誰拿到票最多



how about regression?

Uniform Blending for Regression

$$G(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T g_t(\mathbf{x})$$

classifier 結果加總 取平均

- same g_t (autocracy):
as good as one single g_t
- very different g_t (**diversity + democracy**): *outliner 互相 cancel*
some $g_t(\mathbf{x}) > f(\mathbf{x})$, some $g_t(\mathbf{x}) < f(\mathbf{x})$
 \Rightarrow average **could be** more accurate than individual

diverse hypotheses:

even simple uniform blending
can be better than any single hypothesis

Theoretical Analysis of Uniform Blending

$$G(\underline{\underline{x}}) = \frac{1}{T} \sum_{t=1}^T g_t(\underline{\underline{x}}) \quad \text{先把 } \underline{\underline{x}} \text{ 不写}$$

square error

$$\begin{aligned}
 \text{avg} ((g_t(x) - f(x))^2) &= \text{avg} (g_t^2 - 2g_t f + f^2) \quad G = \text{avg}(g_t) \\
 &= \text{avg}(g_t^2) - 2Gf + f^2 \quad \text{整理} \\
 &= \text{avg}(g_t^2) - \underline{\underline{G^2}} + (G - f)^2 \\
 &= \text{avg}(g_t^2) - \underline{\underline{2G^2}} + \underline{\underline{G^2}} + (G - f)^2 \\
 &= \text{avg}(g_t^2) - \underline{\underline{2g_t G}} + \underline{\underline{G^2}} + (G - f)^2 \\
 &= \text{avg}((g_t - G)^2) + (G - f)^2
 \end{aligned}$$

隨便选的 g_t

$$\begin{aligned}
 \text{avg} (\underline{\underline{E_{\text{out}}(g_t)}}) &= \text{avg} (\mathbb{E}((g_t - G)^2)) + E_{\text{out}}(G) \\
 &\geq \text{必定是正值} + E_{\text{out}}(G)
 \end{aligned}$$

bound by
 $E_{\text{out}}(g_t)$

Questions?

Linear Blending 級不同的票數.

linear blending: known g_t , each to be given α_t ballot

$$G(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^T \alpha_t \cdot g_t(\mathbf{x}) \right) \text{ with } \alpha_t \geq 0$$

票數 \Rightarrow linear combination.
 α_t ≥ 0

computing 'good' α_t :

$$\min_{\alpha_t \geq 0} E_{in}(\alpha)$$

票數 > 0

linear blending for regression

$$\min_{\alpha_t \geq 0} \frac{1}{N} \sum_{n=1}^N \left(y_n - \sum_{t=1}^T \alpha_t g_t(\mathbf{x}_n) \right)^2$$

square error

LinReg + transformation

$$\min_{w_i} \frac{1}{N} \sum_{n=1}^N \left(y_n - \sum_{i=1}^d w_i \phi_i(\mathbf{x}_n) \right)^2$$

等同於 ϕ ?

linear blending = LinModel + hypotheses as transform + constraints

$$\alpha_t \geq 0$$

Constraint on α_t

linear blending = LinModel + hypotheses as transform + constraints:

$$\min_{\alpha_t \geq 0} \quad \frac{1}{N} \sum_{n=1}^N \text{err} \left(y_n, \sum_{t=1}^T \alpha_t g_t(\mathbf{x}_n) \right)$$

linear blending for binary classification

票数 < 0 也不会怎样. \Rightarrow 反过来用

$$\text{if } \underline{\alpha_t < 0} \implies \alpha_t g_t(\mathbf{x}) = |\alpha_t| (-g_t(\mathbf{x}))$$

- negative α_t for $g_t \equiv$ positive $|\alpha_t|$ for $-g_t$
- if you have a stock up/down classifier with 99% error, tell me!
:-)

in practice, often

linear blending = LinModel + hypotheses as transform $\cancel{+ \text{constraints}}$

$\alpha > 0$, 不需要

Linear Blending versus Selection

in practice, often

$$\underline{g_1 \in \mathcal{H}_1, g_2 \in \mathcal{H}_2, \dots, g_T \in \mathcal{H}_T}$$

by minimum E_{in}

- recall: **selection by minimum E_{in}** *dvc 会爆掉*
— best of best, paying $d_{\text{VC}} \left(\bigcup_{t=1}^T \mathcal{H}_t \right)$
- recall: linear blending includes selection as special case
— by setting $\alpha_t = \underline{\llbracket E_{\text{val}}(g_t^-) \text{ smallest} \rrbracket}$
- complexity price of linear blending with E_{in} (aggregation of best):

$$\geq d_{\text{VC}} \left(\bigcup_{t=1}^T \mathcal{H}_t \right)$$

用 E_{in} 做 aggregation? \Rightarrow 複雜度太大

\Rightarrow 不建議用 E_{in} 选.

like selection, blending practically done with
(E_{val} instead of E_{in}) + (g_t^- from minimum E_{train})

用 g^- 找 α , 但最後還是用 g 來訓練

Any Blending

Given $g_1^-, g_2^-, \dots, g_T^-$ from D_{train} , transform (\mathbf{x}_n, y_n) in D_{val} to
 $\mathbf{z}_n = \Phi^-(\mathbf{x}_n, y_n)$, where $\Phi^-(\mathbf{x}) = (g_1^-(\mathbf{x}), \dots, g_T^-(\mathbf{x}))$

Linear Blending

- ① compute α
 $= \text{LinearModel}(\{(\mathbf{z}_n, y_n)\})$
- ② return $G_{LINB}(\mathbf{x}) = \text{LinearHypothesis}_{\alpha}(\Phi(\mathbf{x}))$

where $\Phi(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_T(\mathbf{x}))$

Any Blending (Stacking)

- ① compute \tilde{g} \downarrow 不限於 Linear
 $= \text{AnyModel}(\{(\mathbf{z}_n, y_n)\})$
- ② return $G_{ANYB}(\mathbf{x}) = \tilde{g}(\Phi(\mathbf{x}))$,

any blending:

- **powerful**, achieves conditional blending
- but **danger of overfitting**, as always :-)

Blending in Practice



(Chen et al., A linear ensemble of individual and blended
models for music rating prediction, 2012)

KDDCup 2011 Track 1: World Champion Solution by NTU

- validation set blending: a special **any blending** model

$$E_{\text{test}} \text{ (squared): } \underline{\underline{519.45}} \implies 456.24$$

—helped **secure the lead** in last two weeks

- test set blending: linear blending using \tilde{E}_{test}

再做 \rightarrow R Linear blending

$$E_{\text{test}} \text{ (squared): } 456.24 \implies 442.06$$

—helped **turn the tables** in last hour

blending ‘useful’ in practice,
despite the computational burden

Questions?

What We Have Done

blending: aggregate **after getting g_t** ;

learning: aggregate **as well as getting g_t** *g_t How to get?*

aggregation type	blending	learning
公平 uniform	voting/averaging	?
不公平 non-uniform	linear	?
條件 conditional	stacking	?

learning g_t for uniform aggregation: diversity important 不一樣的 g_t

- diversity by different models: $g_1 \in \mathcal{H}_1, g_2 \in \mathcal{H}_2, \dots, g_T \in \mathcal{H}_T$
- diversity by different parameters: GD with $\eta = 0.001, 0.01, \dots, 10$
- diversity by algorithmic randomness:
random PLA with different random seeds
- diversity by data randomness: \leftarrow sample data.
within-cross-validation hypotheses g_v^-

next: diversity by data randomness without g^-

Bootstrap Aggregation

bootstrapping

bootstrap sample $\tilde{\mathcal{D}}_t$: re-sample N examples from \mathcal{D} uniformly with replacement—can also use arbitrary N' instead of original N

- $\tilde{\mathcal{D}}_t \neq \mathcal{D}$ in general (diverse)
- $\tilde{\mathcal{D}}_t$ ‘acts like’ \mathcal{D} statistically (g_t ‘acts like’ good g)
- each g_t similarly good/bad (uniform)

bootstrap aggregation

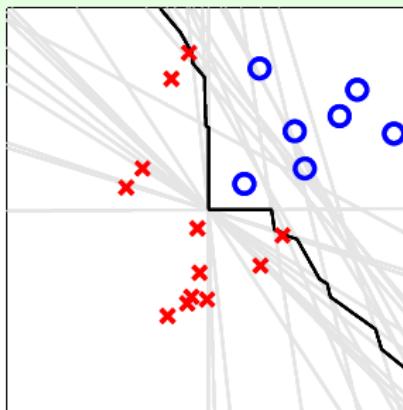
iterative process that for $t = 1, 2, \dots, T$

- ① request size- N' data $\tilde{\mathcal{D}}_t$ from bootstrapping
- ② obtain g_t by $\mathcal{A}(\tilde{\mathcal{D}}_t)$ \leftarrow 用 $\tilde{\mathcal{D}}_t$ 得到 g_t .
 $G = \text{Uniform}(\{g_t\})$

bootstrap aggregation (BAGging):

a simple meta algorithm on top of base algorithm

Bagging on Top of (Modified) PLA in Action



$$T_{\text{BAG}} = 25$$

- very diverse g_t from bagging
- proper non-linear boundary after aggregating binary classifiers

bagging works reasonably well if base
algorithm sensitive to data randomness

diversity才高

Questions?

Apple Recognition Problem

- is this a picture of an apple?
- say, want to teach a class of **6 year olds**
- gather photos under CC-BY-2.0 license on Flickr
(thanks to the authors below!)

(APAL stands for Apple and Pear Australia Ltd)



Dan Foy

<https://flic.kr/p/jNQ55>



APAL

<https://flic.kr/p/jzP1VB>



adrianbartel

<https://flic.kr/p/bdy2hZ>



ANdrzej ch.

<https://flic.kr/p/51DKA8>



Stuart Webster

<https://flic.kr/p/9C3Ybd>



nachans

<https://flic.kr/p/9XD7Ag>



APAL

<https://flic.kr/p/jzRe4u>



Jo Jakeman

<https://flic.kr/p/7jwtGp>



APAL

<https://flic.kr/p/jzPYNr>



APAL

<https://flic.kr/p/jzScif>

Apple Recognition Problem

- is this a picture of an apple?
- say, want to teach a class of **6 year olds**
- gather photos under CC-BY-2.0 license on Flickr
(thanks to the authors below!)



Mr. Roboto.

<https://flic.kr/p/i5BN85>



Richard North

<https://flic.kr/p/bHhPkB>



Richard North

<https://flic.kr/p/d8tGou>



Emilian Vicol

<https://flic.kr/p/bpmGXW>



Nathaniel Queen

<https://flic.kr/p/pZv1Mf>



Crystal

<https://flic.kr/p/kaPYp>



jf686

<https://flic.kr/p/6vjRFH>



skyseeker

<https://flic.kr/p/2MynV>



Janet Hudson

<https://flic.kr/p/7QDBbm>



Rennett Stowe

<https://flic.kr/p/agmnrk>

Our Fruit Class Begins

- Teacher: Please look at the pictures of apples and non-apples below. Based on those pictures, how would you describe an apple? Michael?
- Michael: I think apples are **circular**.

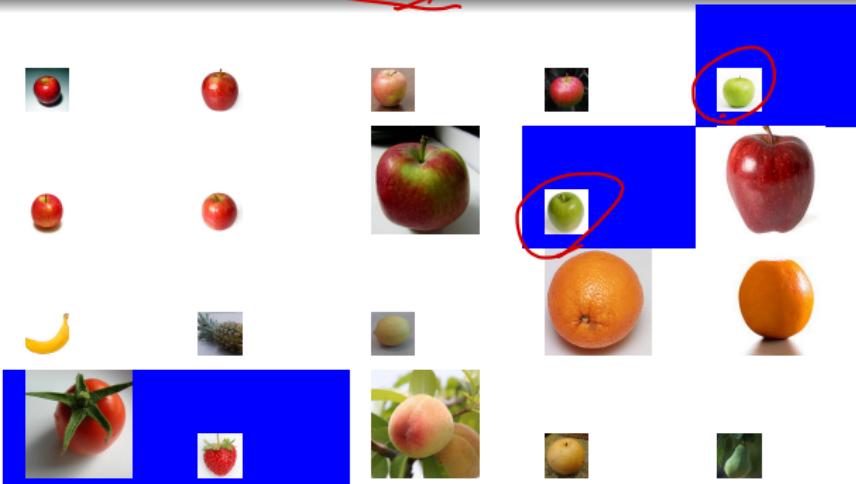
is apple

not apple

(Class): Apples are **circular**.

Our Fruit Class Continues

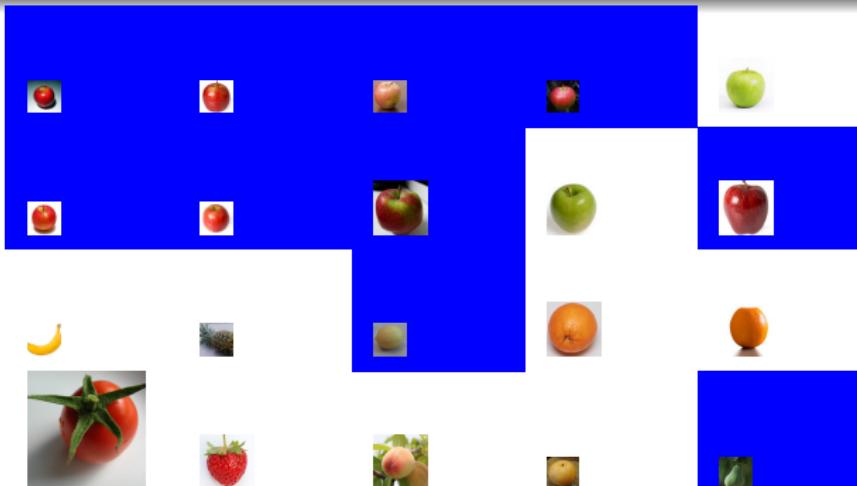
- Teacher: Being circular is a good feature for the apples. However, if you only say circular, you could make several mistakes. What else can we say for an apple? Tina?
- Tina: It looks like apples are **red**.



(Class): Apples are somewhat **circular** and somewhat **red**.

Our Fruit Class Continues More

- Teacher: Yes. Many apples are red. However, you could still make mistakes based on circular and red. Do you have any other suggestions, Joey?
- Joey: Apples could also be **green**.



(Class): Apples are somewhat **circular** and somewhat **red** and possibly **green**.

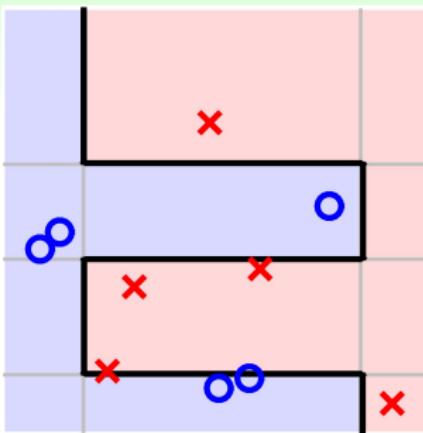
Our Fruit Class Ends

- Teacher: Yes. It seems that apples might be circular, red, green. But you may confuse them with tomatoes or peaches, right? Any more suggestions, Jessica?
- Jessica: Apples have **stems** at the top.



(Class): Apples are somewhat **circular**, somewhat **red**, possibly **green**, and may have **stems** at the top.

Motivation



- students: simple hypotheses g_t (like vertical/horizontal lines)
- (Class): sophisticated hypothesis G (like black curve)
- Teacher: a tactic learning algorithm that **directs the students to focus on key examples**

next: the '**math**' of such an algorithm

Questions?

Bootstrapping as Re-weighting Process

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_3, y_3), (\mathbf{x}_4, y_4)\}$$

bootstrap $\Rightarrow \tilde{\mathcal{D}}_t = \{(\mathbf{x}_1, y_1), (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), (\mathbf{x}_4, y_4)\}$

重複

weighted E_{in} on \mathcal{D}

$$E_{\text{in}}^{\mathbf{u}}(h) = \frac{1}{4} \sum_{n=1}^4 u_n^{(t)} \cdot [\![y_n \neq h(\mathbf{x}_n)]\!]$$

每輪都不同

$$\left\{ \begin{array}{l} (\mathbf{x}_1, y_1), u_1 = 2 \\ (\mathbf{x}_2, y_2), u_2 = 1 \\ (\mathbf{x}_3, y_3), u_3 = 0 \\ (\mathbf{x}_4, y_4), u_4 = 1 \end{array} \right.$$

E_{in} on $\tilde{\mathcal{D}}_t$

$$E_{\text{in}}^{0/1}(h) = \frac{1}{4} \sum_{(\mathbf{x}, y) \in \tilde{\mathcal{D}}_t} [\![y \neq h(\mathbf{x})]\!]$$

一樣

$$\begin{aligned} & (\mathbf{x}_1, y_1), (\mathbf{x}_1, y_1) \\ & (\mathbf{x}_2, y_2) \\ & (\mathbf{x}_4, y_4) \end{aligned}$$

出現次數作為權重，用 bootstrap 得到 u ，再 minimize
each diverse g_t in bagging:

by minimizing bootstrap-weighted error

B.

Weighted Base Algorithm

minimize (regularized)

$$E_{\text{in}}^{\mathbf{u}}(h) = \frac{1}{N} \sum_{n=1}^N \underline{u_n} \cdot \text{err}(y_n, h(\mathbf{x}_n))$$

e.g. logistic regression

$$E_{\text{in}}^{\mathbf{u}} \propto \sum_{n=1}^N \underline{u_n} \text{err}_{\text{CE}} \text{ by SGD}$$

\Leftrightarrow sample (\mathbf{x}_n, y_n) with
probability proportional to $\underline{u_n}$

example-weighted learning:

modified error function, optimized by sampling/re-derivation/...

Re-weighting for More Diverse Hypothesis

'improving' bagging for binary classification: 如何決定 $u_n^{(t)}$
 how to re-weight for **more diverse hypotheses**?

$$g_t \leftarrow \operatorname{argmin}_{h \in \mathcal{H}} \left(\sum_{n=1}^N u_n^{(t)} \llbracket y_n \neq h(\mathbf{x}_n) \rrbracket \right)$$

$$g_{t+1} \leftarrow \operatorname{argmin}_{h \in \mathcal{H}} \left(\sum_{n=1}^N u_n^{(t+1)} \llbracket y_n \neq h(\mathbf{x}_n) \rrbracket \right)$$

if g_t 'not good' for $\mathbf{u}^{(t+1)}$ ↪ 选 $\mathbf{u}_n^{(t+1)}$ s.t. g_t 在 g_{t+1} 球表現很糟
 $\Rightarrow g_t$ -like hypotheses not returned as g_{t+1}
 $\Rightarrow g_{t+1}$ diverse from g_t

idea: **construct $\mathbf{u}^{(t+1)}$ to make g_t random-like**

$$\frac{\sum_{n=1}^N u_n^{(t+1)} \llbracket y_n \neq g_t(\mathbf{x}_n) \rrbracket}{\sum_{n=1}^N u_n^{(t+1)}} = \frac{1}{2}$$

在 $\mathbf{u}_n^{(t+1)}$ 量 小 g_t 去硬幣
 binary classifier.

Optimal' Re-weighting 犯錯的桌

want: $\frac{\sum_{n=1}^N u_n^{(t+1)} [y_n \neq g_t(\mathbf{x}_n)]}{\sum_{n=1}^N u_n^{(t+1)}} = \frac{\blacksquare_{t+1}}{\blacksquare_{t+1} + \bullet_{t+1}} = \frac{1}{2}$, where

$$\blacksquare_{t+1} = \sum_{n=1}^N u_n^{(t+1)} [y_n \neq g_t(\mathbf{x}_n)], \bullet_{t+1} = \sum_{n=1}^N u_n^{(t+1)} [y_n = g_t(\mathbf{x}_n)]$$

犯錯

沒犯錯

- need: $\underbrace{\text{(total } u_n^{(t+1)} \text{ of incorrect)}}_{\blacksquare_{t+1}} = \underbrace{\text{(total } u_n^{(t+1)} \text{ of correct)}}_{\bullet_{t+1}} \Rightarrow \text{就 } \frac{1}{2} \text{ 了!}$
- one possibility by **re-scaling (multiplying) weights**, if

$$(\text{total } u_n^{(t)} \text{ of incorrect}) = 1126;$$

$$(\text{weighted incorrect rate}) = \frac{1126}{7337}$$

$$\text{incorrect: } u_n^{(t+1)} \leftarrow u_n^{(t)} \cdot 6211$$

$$(\text{total } u_n^{(t)} \text{ of correct}) = 6211;$$

$$(\text{weighted correct rate}) = \frac{6211}{7337}$$

$$\text{correct: } u_n^{(t+1)} \leftarrow u_n^{(t)} \cdot 1126$$

'optimal' re-weighting under weighted incorrect rate ϵ_t :

multiply **incorrect** $\propto (1 - \epsilon_t)$; multiply **correct** $\propto \epsilon_t$

Questions?

Scaling Factor

'optimal' re-weighting: let $\epsilon_t = \frac{\sum_{n=1}^N u_n^{(t)} [y_n \neq g_t(\mathbf{x}_n)]}{\sum_{n=1}^N u_n^{(t)}}$,

multiply incorrect $\propto (1 - \epsilon_t)$; multiply correct $\propto \epsilon_t$

define scaling factor $\diamond_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$

incorrect

correct

incorrect

correct



- equivalent to optimal re-weighting

- $\diamond_t \geq 1$ iff $\epsilon_t \leq \frac{1}{2} \Rightarrow$ incorrect 放大, correct 缩小

—physical meaning: scale up incorrect; scale down correct
—like what Teacher does 強調 mistake.

scaling-up incorrect examples
leads to diverse hypotheses

A Preliminary Algorithm

$\mathbf{u}^{(1)} = ?$ *U的initialised value.*

for $t = 1, 2, \dots, T$

- ① obtain g_t by $\mathcal{A}(\mathcal{D}, \mathbf{u}^{(t)})$,
where \mathcal{A} tries to minimize $\mathbf{u}^{(t)}$ -weighted 0/1 error
- ② update $\mathbf{u}^{(t)}$ to $\mathbf{u}^{(t+1)}$ by $\diamond_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$,
where ϵ_t = weighted error (incorrect) rate of g_t

return $G(\mathbf{x}) = ?$

- want g_1 'best' for E_{in} : $\underline{\underline{u_n^{(1)} = \frac{1}{N}}}$ *good ↓ bad ↓*
- $G(\mathbf{x})$:
 - uniform? but g_2 very bad for E_{in} (why? :-)
 - linear, non-linear? **as you wish**

一追決定g, 一追決定d.

next: a special algorithm to aggregate
linearly on the fly with theoretical guarantee

Linear Aggregation on the Fly

$$\mathbf{u}^{(1)} = [\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}] \leftarrow \text{initial value}$$

for $t = 1, 2, \dots, T$

① obtain g_t by $\mathcal{A}(\mathcal{D}, \mathbf{u}^{(t)})$, where ...

② update $\mathbf{u}^{(t)}$ to $\mathbf{u}^{(t+1)}$ by $\diamond_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$, where ...

③ compute $\alpha_t = \ln(\diamond_t)$

return $G(\mathbf{x}) = \text{sign}\left(\sum_{t=1}^T \alpha_t g_t(\mathbf{x})\right)$

票數.

$\diamond_t \uparrow$, α_t 也大.

- wish: large α_t for 'good' $g_t \Leftarrow \alpha_t = \text{monotonic}(\diamond_t)$

- will take $\alpha_t = \ln(\diamond_t)$

- $\epsilon_t = \frac{1}{2} \Rightarrow \diamond_t = 1 \Rightarrow \alpha_t = 0$ (bad g_t zero weight)

- $\epsilon_t = 0 \Rightarrow \diamond_t = \infty \Rightarrow \alpha_t = \infty$ (super g_t superior weight)

正確率 100%, 独立

決定 g .

Adaptive Boosting = weak base learning algorithm \mathcal{A} (Student)

+ optimal re-weighting factor \diamond_t (Teacher) \diamond_t

+ 'magic' linear aggregation α_t (Class) 決定 α_t .

Adaptive Boosting (AdaBoost) Algorithm

$\mathbf{u}^{(1)} = [\frac{1}{N}, \frac{1}{N}, \dots, \frac{1}{N}]$ ← 所有步的重要性都一樣。完整 AdaBoost.
for $t = 1, 2, \dots, T$

$$\epsilon_t < \frac{1}{2}$$

- ① obtain g_t by $\mathcal{A}(\mathcal{D}, \mathbf{u}^{(t)})$,
where \mathcal{A} tries to minimize $\mathbf{u}^{(t)}$ -weighted 0/1 error
- ② update $\mathbf{u}^{(t)}$ to $\mathbf{u}^{(t+1)}$ by

$$\begin{aligned} \llbracket y_n \neq g_t(\mathbf{x}_n) \rrbracket \text{ (incorrect examples): } u_n^{(t+1)} &\leftarrow u_n^{(t)} \cdot \diamond_t \\ \llbracket y_n = g_t(\mathbf{x}_n) \rrbracket \text{ (correct examples): } u_n^{(t+1)} &\leftarrow u_n^{(t)} / \diamond_t \end{aligned}$$

where $\diamond_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$ and $\epsilon_t = \frac{\sum_{n=1}^N u_n^{(t)} \llbracket y_n \neq g_t(\mathbf{x}_n) \rrbracket}{\sum_{n=1}^N u_n^{(t)}} \leftarrow \text{By } g_t \text{ 错误率}$

- ③ compute $\alpha_t = \ln(\diamond_t)$

return $G(\mathbf{x}) = \text{sign} \left(\sum_{t=1}^T \alpha_t g_t(\mathbf{x}) \right)$ ← linear aggregation.

AdaBoost: provable **boosting property**

Theoretical Guarantee of AdaBoost

- From VC bound

$$E_{\text{out}}(G) \leq E_{\text{in}}(G) + O\left(\sqrt{\underbrace{O(d_{VC}(\mathcal{H}) \cdot T \log T)}_{d_{VC} \text{ of all possible } G} \cdot \frac{\log N}{N}}\right)$$

模型 complexity
of data.

- 5g. # T log T 很小,

- first term can be small:**

$E_{\text{in}}(G) = 0$ after $T = O(\log N)$ iterations if $\epsilon_t \leq \epsilon < \frac{1}{2}$ always

- second term can be small:** 很快把 $E_{\text{in}}(G) = 0$

overall d_{VC} grows “slowly” with T

boosting view of AdaBoost:

比弱略好就好

if \mathcal{A} is weak but always slightly better than random ($\epsilon_t < \epsilon < \frac{1}{2}$),
 then (AdaBoost+ \mathcal{A}) can be strong ($E_{\text{in}} = 0$ and E_{out} small)

Questions?

Decision Stump *← algorithm A.*

want: a '**weak**' base learning algorithm *A*

that minimizes $E_{\text{in}}^{\text{u}}(h) = \frac{1}{N} \sum_{n=1}^N u_n \cdot [y_n \neq h(\mathbf{x}_n)]$ *a little bit*

example 重要性

a popular choice: decision stump

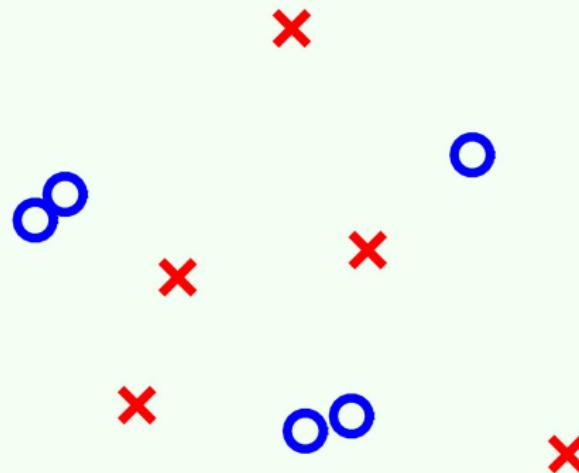
- $S = \{+1, -1\}$ 找所有參數的組合
- $h_{s,i,\theta}(\mathbf{x}) = s \cdot \text{sign}(x_i - \theta)$ threshold.
ith feature.
- positive and negative rays on some feature: three parameters
(feature *i*, threshold *θ*, direction *s*) 真的能把 E_{in} 做到很小
- physical meaning: vertical/horizontal lines in 2D
- efficient to optimize: $O(d \cdot N \log N)$ time

decision stump model:

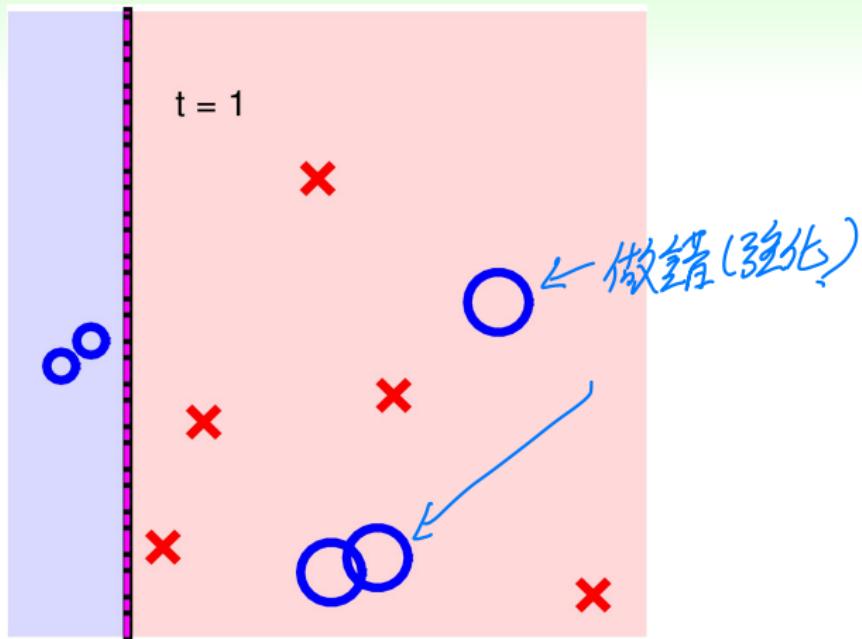
allows efficient minimization of E_{in}^{u}
but perhaps **too weak to work by itself**

A Simple Data Set

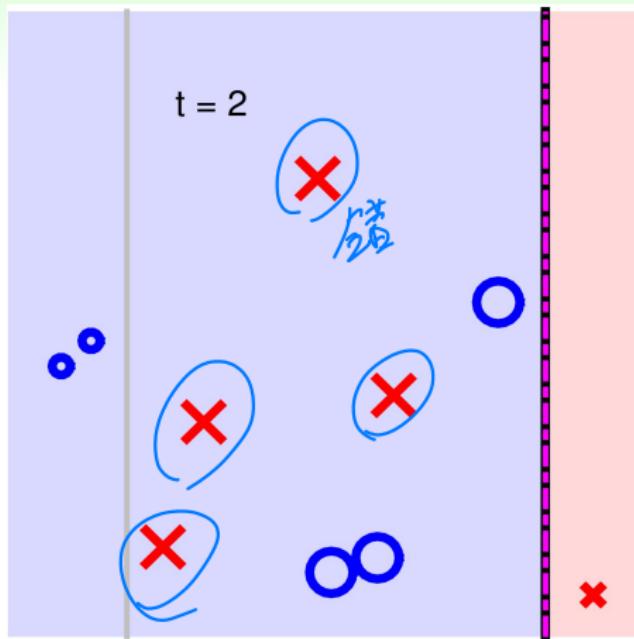
initially



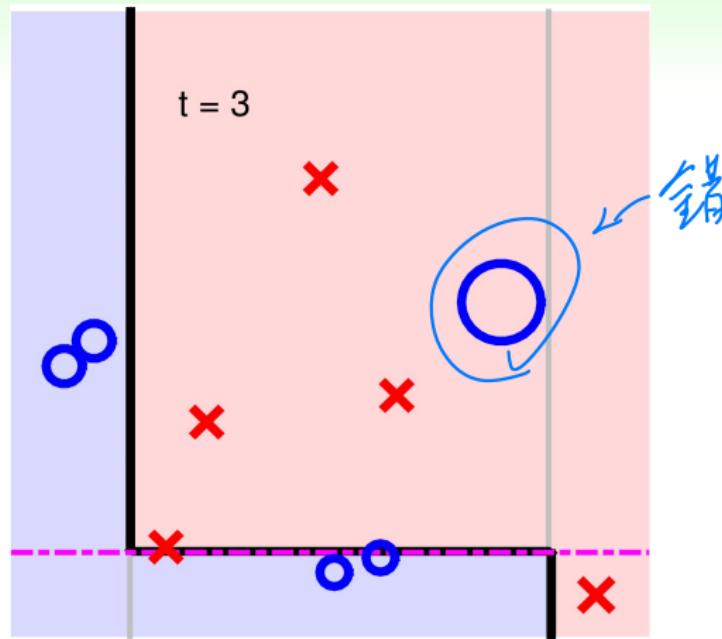
A Simple Data Set



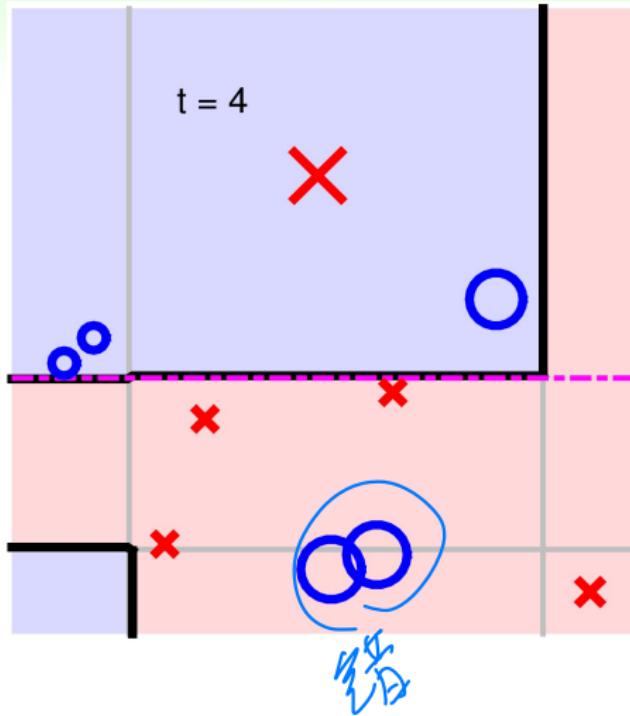
A Simple Data Set



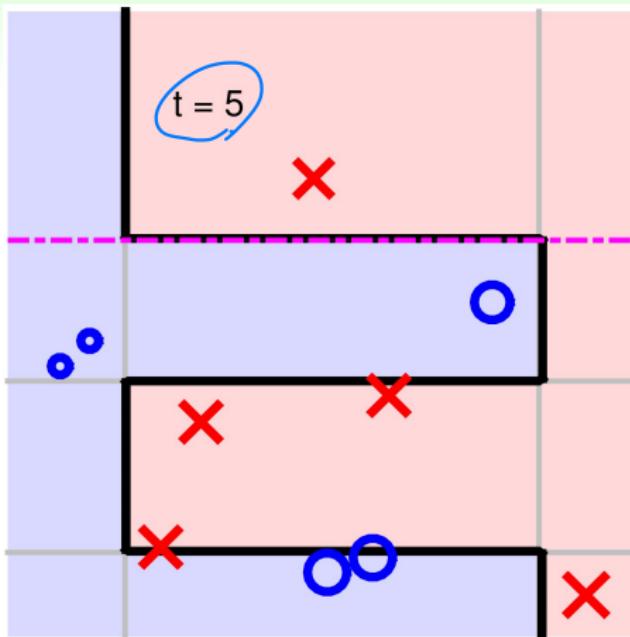
A Simple Data Set



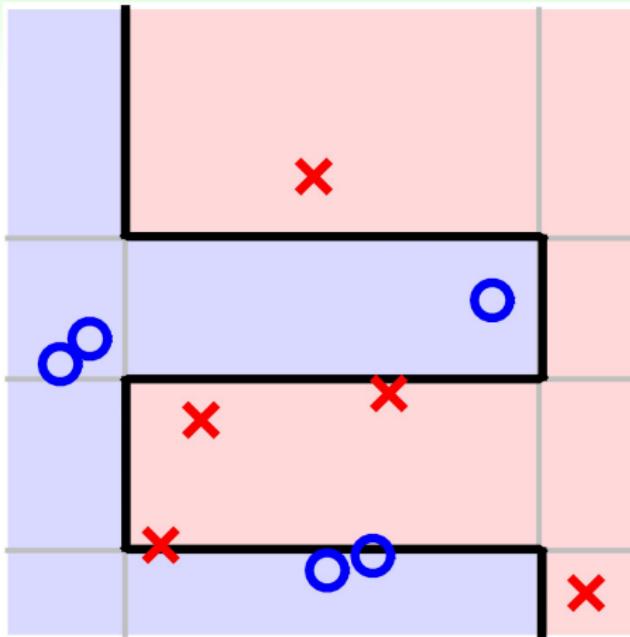
A Simple Data Set



A Simple Data Set



A Simple Data Set



‘Teacher’-like algorithm works!

Putting Everything Together

Gradient Boosted Decision Tree (GBDT)

$$s_1 = s_2 = \dots = s_N = 0$$

for $t = 1, 2, \dots, T$

- ① obtain g_t by $\mathcal{A}(\{(x_n, y_n - s_n)\})$ where \mathcal{A} is a (squared-error) regression algorithm

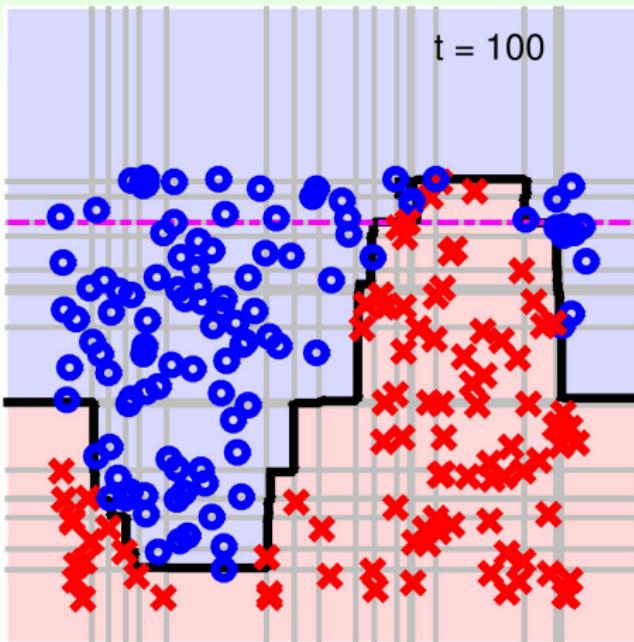
—**how about sampled and pruned C&RT?**

- ② compute $\alpha_t = \text{OneVarLinearRegression}(\{(g_t(x_n), y_n - s_n)\})$
- ③ update $s_n \leftarrow s_n + \alpha_t g_t(x_n)$

return $G(\mathbf{x}) = \sum_{t=1}^T \alpha_t g_t(\mathbf{x})$

GBDT: ‘regression sibling’ of AdaBoost-DTree
—**popular in practice**

A Complicated Data Set



AdaBoost-Stump: **non-linear yet efficient**

AdaBoost-Stump in Application



original picture by F.U.S.I.A. assistant and derivative work by Sylenius via Wikimedia Commons

The World's First 'Real-Time' Face Detection Program

- AdaBoost-Stump as core model: linear aggregation of key patches selected out of 162,336 possibilities in 24x24 images
 - feature selection achieved through AdaBoost-Stump
- modified linear aggregation G to rule out non-face earlier
 - efficiency achieved through modified linear aggregation

AdaBoost-Stump:

efficient feature selection and aggregation

Questions?

Summary

① Combining Predictive Features: Aggregation Models

Lecture 12: Bagging and Boosting

- Uniform Blending
 - diverse hypotheses, 'one vote, one value'**
- Linear and Any Blending
 - two-level learning with hypotheses as transform**
- Bagging (Bootstrap Aggregation)
 - bootstrapping for diverse hypotheses**
- Motivation of Boosting
 - aggregate weak hypotheses for strength**
- Diversity by Re-weighting
 - scale up incorrect, scale down correct**
- Adaptive Boosting Algorithm
 - two heads are better than one, theoretically**
- Adaptive Boosting in Action
 - AdaBoost-Stump useful and efficient**