自適應增強 AdaBoost

國立東華大學電機工程學系 楊哲旻

Outline

1. 提升
2. 集成學習
3. 重新加權訓練集
4. 演算法
5. 均勻與非均勻權重
6. 範例

機器學習 – 自適應增強 01. 提升

提升(Boosting)

如果機器學習算法可在訓練集上獲得誤差小於50%的分類器。提升法後可獲得0%誤差的分類器

1. 模型訓練後獲得第一個分類器 $f_1(x)$
2. 找到第二個分類器 $f_2(x)$ 能幫助 $f_1(x)$更好的分類
   - 如果 $f_2(x)$ 與 $f_1(x)$相似，則沒有任何幫助
   - 希望 $f_2(x)$ 與 $f_1(x)$為互補關係
3. 獲得第二個分類器 $f_2 x$
4. …… 最後，合併所有分類器

※ 註解：分類器是順序學習的

機器學習 – 自適應增強　01. 集成學習

集成學習 (Ensemble learning)

Q1: 如何獲得不同的分類器？

A1: 訓練不同的訓練集

Q2: 如何得到不同的訓練集？

A2-1: 重新取樣你的訓練集以得到一新集合，例如：裝袋法(Bagging)

A2-2: 重新加權你的訓練集以得到一新集合，例如：提升法(Boosting)

只需要改變損失函數就可以實現

$$(x^1, \hat{y}^1, u^1) \qquad u^1 = \cancel{1}\ 0.4$$

$$(x^2, \hat{y}^2, u^2) \qquad u^2 = \cancel{1}\ 2.1$$

$$\vdots \qquad\qquad \vdots$$

$$(x^n, \hat{y}^n, u^n) \qquad u^n = \cancel{1}\ 1.3$$

$$\boxed{L(f) = \sum_n l(f(x^n), \hat{y}^n)}$$

$$\boxed{L(f) = \sum_n u^n l(f(x^n), \hat{y}^n)}$$

機器學習 – 自適應增強 01. 集成學習

集成學習 (Ensemble learning)

想法：對 $f_1(x)$產生能訓練失敗的新訓練集，再進行 $f_2(x)$ 的訓練

如何得到能讓 $f_1(x)$訓練失敗的新訓練集?

ε_1 為訓練集在$f_1(x)$ 上的誤差

$$\varepsilon_1 = \frac{\sum_n u_1^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_1} \qquad Z_1 = \sum_n u_1^n \qquad \varepsilon_1 < 0.5$$

改變樣本的權重 u_1^n 至 u_2^n $\qquad \dfrac{\sum_n u_2^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_2} = 0.5$

根據新的權重 u_2^n之訓練集再次訓練得到 $f_2(x)$

機器學習 – 自適應增強 02.重新加權訓練集

重新加權訓練集 (Re-weighting Training Data)

如何得到能讓 $f_1(x)$訓練失敗的新訓練集?

$(x^1, \hat{y}^1, u^1)$	$u^1 = 1$	✓	$u^1 = 1/\sqrt{3}$	✓
$(x^2, \hat{y}^2, u^2)$	$u^2 = 1$	✗	$u^2 = \sqrt{3}$	✗
$(x^3, \hat{y}^3, u^3)$	$u^3 = 1$	✓	$u^3 = 1/\sqrt{3}$	✓
$(x^4, \hat{y}^4, u^4)$	$u^4 = 1$	✓	$u^4 = 1/\sqrt{3}$	✓
	$\boxed{\varepsilon_1 = 0.25}$		$\boxed{\varepsilon_1 = 0.5}$	
	$f_1(x)$		$f_1(x)$	$f_2(x)$

機器學習 – 自適應增強　02.重新加權訓練集

重新加權訓練集 (Re-weighting Training Data)

如何得到能讓 $f_1(x)$訓練失敗的新訓練集?

- 若 x^n 被分類錯誤 $f_1(f_1(x^n) \neq \hat{y}^n)$

$u_2^n \leftarrow u_1^n$ 乘以 d_1　increase

- 若 x^n 被分類正確 $f_1(f_1(x^n) = \hat{y}^n)$

$u_2^n \leftarrow u_1^n$ 除以 d_1　decrease

以新的權重u_2^n的資料來訓練f_2

d_1如何得到？

機器學習 – 自適應增強 02.重新加權訓練集

重新加權訓練集 (Re-weighting Training Data)

$$\varepsilon_1 = \frac{\sum_n u_1^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_1} \qquad Z_1 = \sum_n u_1^n$$

$$\frac{\boxed{\sum_n u_2^n \delta(f_1(x^n) \neq \hat{y}^n)}}{\boxed{Z_2}} = 0.5$$

$f_1(x^n) \neq \hat{y}^n$ $\quad u_2^n \leftarrow u_1^n$ multiplying d_1

$f_1(x^n) = \hat{y}^n$ $\quad u_2^n \leftarrow u_1^n$ devided by d_1

$$= \sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1$$

$$= \sum_n u_2^n = \sum_{f_1(x^n) \neq \hat{y}^n} u_2^n + \sum_{f_1(x^n) = \hat{y}^n} u_2^n = \sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1 + \sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1$$

$$\frac{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1}{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1 + \sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1} = 0.5 \Rightarrow \frac{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1 + \sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1}{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1} = 2$$

機器學習 – 自適應增強 02.重新加權訓練集

重新加權訓練集 (Re-weighting Training Data)

$$\frac{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1 + \sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1}{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1} = 2$$

$$\Rightarrow \sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1 = \sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1$$

$$\Rightarrow d_1 \sum_{f_1(x^n) \neq \hat{y}^n} u_1^n = \frac{1}{d_1} \sum_{f_1(x^n) = \hat{y}^n} u_1^n$$

$$\Rightarrow d_1 Z_1 \varepsilon_1 = \frac{1}{d_1} Z_1 (1 - \varepsilon_1)$$

$$\Rightarrow d_1 = \sqrt{(1 - \varepsilon_1)/\varepsilon_1}$$

機器學習 – 自適應增強 03.演算法

自適應增強演算法 (Algorithm for AdaBoost)

- Giving training data $\{(x^1, \hat{y}^1, u^1), \ldots, (x^n, \hat{y}^n, u^n), \ldots, (x^N, \hat{y}^N, u^N)\}$

$\hat{y} = \pm 1$(Binary classification), $u_1^n = 1$(equal weights)

- For t = 1, …, T:
  - ✓ Training weak classifier $f_t(x)$ with weights $\{u_t^1, \ldots, u_t^N\}$
  - ✓ ε_t is the error rate of $f_t(x)$ with weights$\{u_t^1, \ldots, u_t^N\}$
  - ✓ For n = 1, …, N:
    - ➢ If x^n is misclassified classified by $f_t(x)$: $\boxed{f_t(x^n) \neq \hat{y}^n}$

      $u_{t+1}^n = u_t^n \times d_t = u_t^n \times \exp(\alpha_t)$
    - ➢ Else:

      $u_{t+1}^n = u_t^n / d_t = u_t^n \times \exp(-\alpha_t)$

$$\boxed{u_{t+1}^n = u_t^n \times d_t = u_t^n \times \exp(-\hat{y}^n f_t(x^n) \alpha_t)}$$

$$\boxed{d_t = \sqrt{(1-\varepsilon_t)/\varepsilon_t}}$$

$$\boxed{\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}}$$

機器學習 – 自適應增強 04. 均勻與非均勻權重

自適應增強演算法 (Algorithm for AdaBoost)

最終獲得所有分類器: $f_1(x), \dots, f_t(x), \dots, f_T(x)$

如何合併它們?

- 均勻權重(Uniform weight) :

$\boxed{\alpha_t = ln\sqrt{(1-\varepsilon_t)/\varepsilon_t}}$

$$H(x) = sign(\sum_{t=1}^{T} f_t(x))$$

$\boxed{u_{t+1}^n = u_t^n \times d_t = u_t^n \times \exp(-\hat{y}^n f_t(x^n)\alpha_t)}$

- 非均勻權重(Non-uniform weight) :

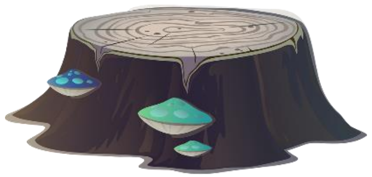
$$H(x) = sign(\sum_{t=1}^{T} \alpha_t f_t(x))$$

錯誤率小 ε_t 給予投票的權重較大

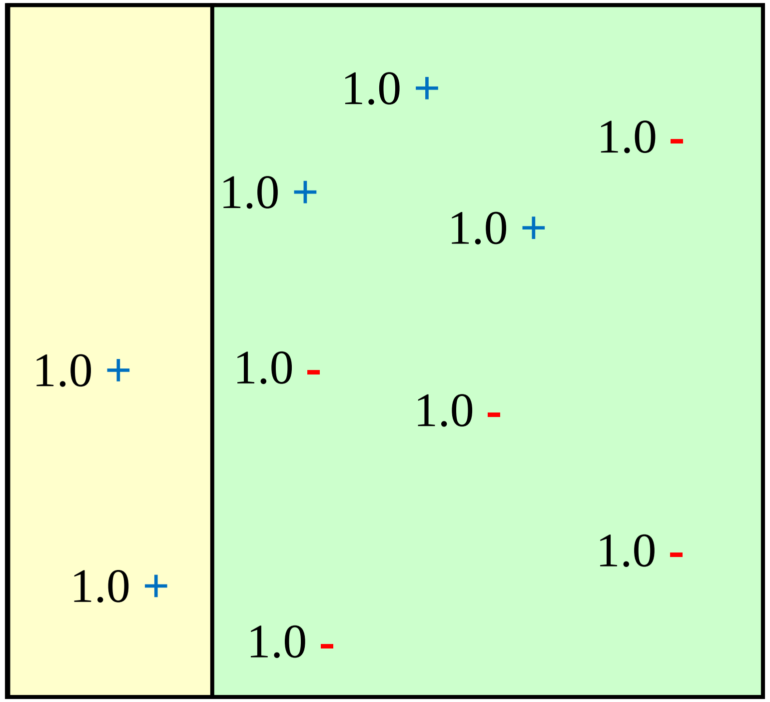
機器學習 – 自適應增強 04. 範例

Toy Example

T = 3, weak classifier = decision stump



t = 1

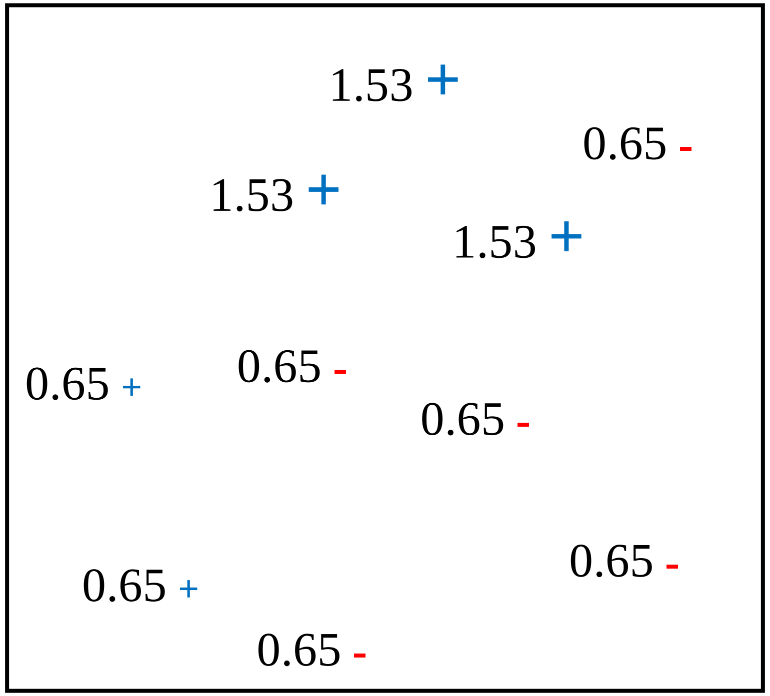


$f_1(x)$

$\varepsilon_1 = 0.30$

$d_1 = 1.53$

$\alpha_1 = 0.42$

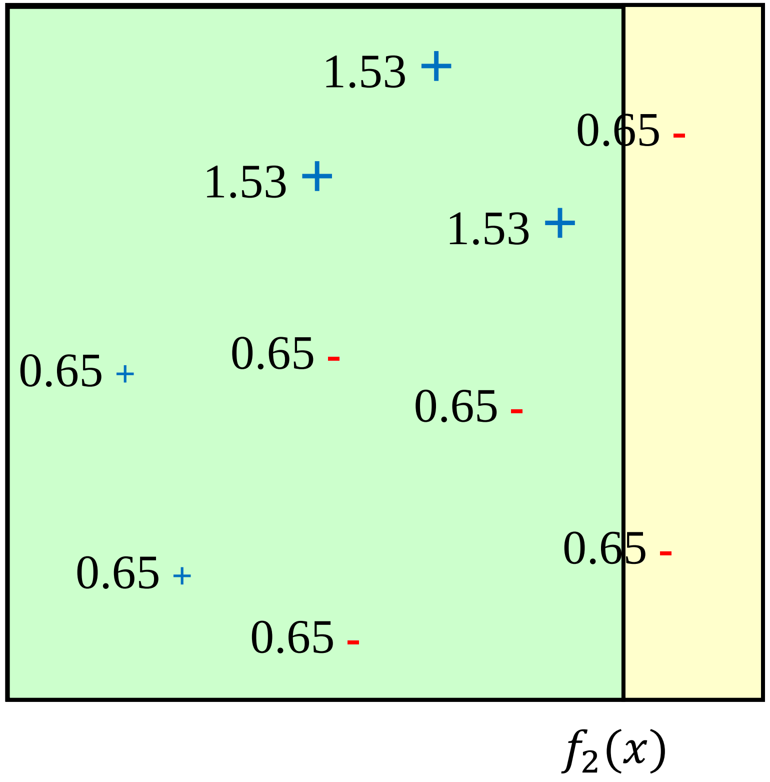


機器學習 – 自適應增強 04. 範例

Toy Example

T = 3, weak classifier = decision stump

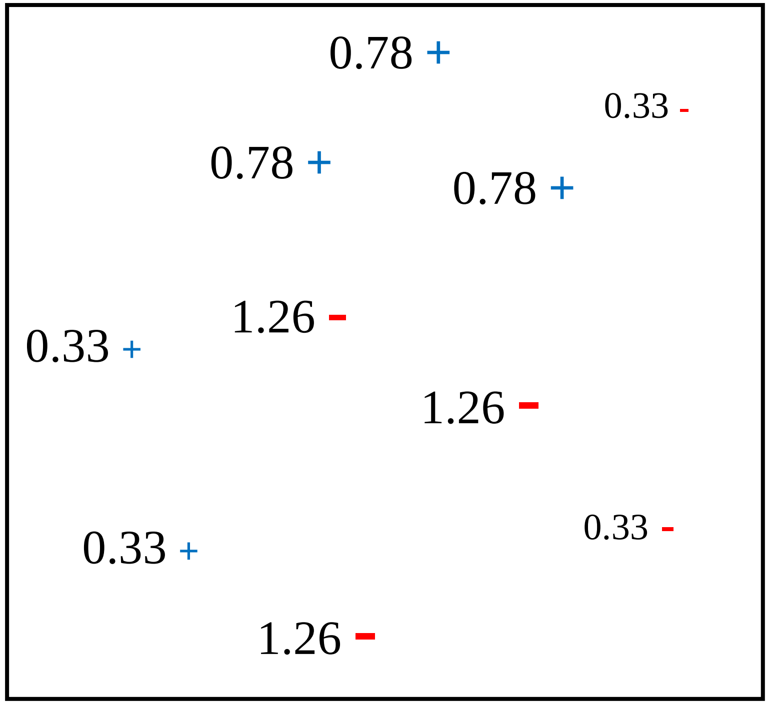
t = 2



$\varepsilon_2 = 0.21$

$d_2 = 1.94$

$\alpha_2 = 0.66$

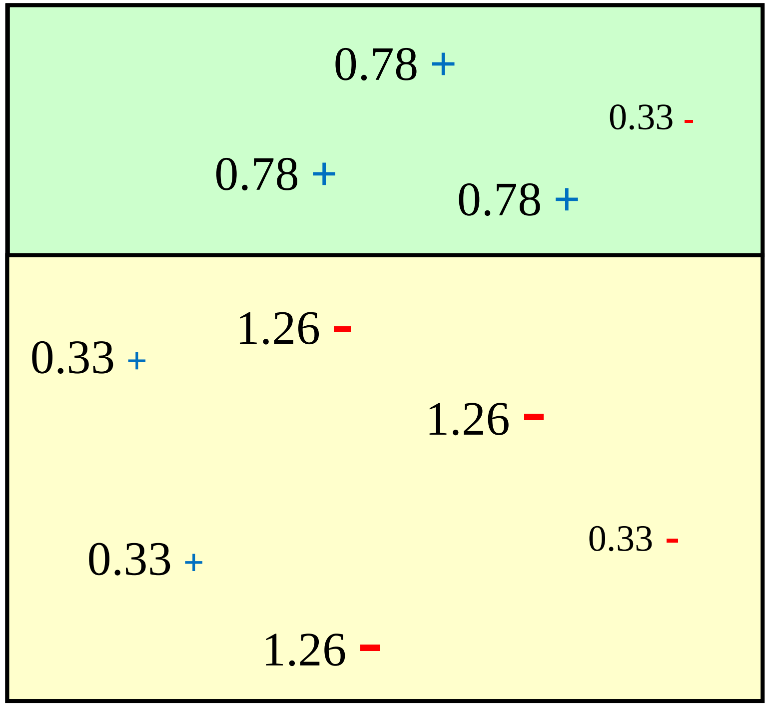


機器學習 – 自適應增強 04. 範例

Toy Example

T = 3, weak classifier = decision stump

t = 3



$f_3(x)$

$\varepsilon_3 = 0.13$

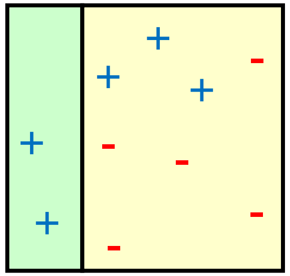
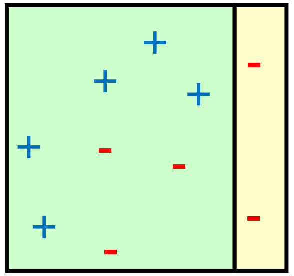
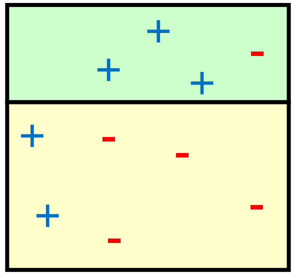
$d_3 = 2.59$

$\alpha_3 = 0.95$

機器學習 – 自適應增強 04. 範例

Toy Example

T = 3, weak classifier = decision stump

$sign(\ 0.42$  $+\ 0.66$  $+\ 0.95$ )

Final Classifier:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t f_t(x))$$