Machine Learning
(機器學習)

Lecture 8: Combatting Overfitting (2)

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**?

Lecture 8: Combatting Overfitting (2)

- Model Selection Problem
- Validation
- Leave-One-Out Cross Validation
- V-Fold Cross Validation

So Many Models Learned

Even Just for Binary Classification . . .

$\mathcal{A} \in$ { PLA, pocket, linear regression, logistic regression}

$\times$

$T \in \{100, 1000, 10000\}$ ← 走幾步

$\times$

$\eta \in \{1, 0.01, 0.0001\}$ ← learning rate

$\times$

$\Phi \in$ { linear, quadratic, poly-10, Legendre-poly-10} ← feature transform

$\times$

$\Omega(\mathbf{w}) \in$ { L2 regularizer, L1 regularizer, symmetry regularizer}

↑ regularizer?

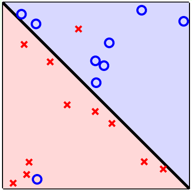
$\times$

$\lambda \in \{0, 0.01, 1\}$ ← λ

太多組合

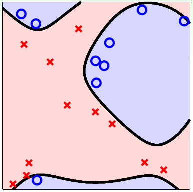
in addition to your **favorite** combination, may need to try other combinations to get a good g

Model Selection Problem



$\mathcal{H}_1$

which one do you prefer? :-)



$\mathcal{H}_2$

- given: M models $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_M$, each with corresponding algorithm $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_M$

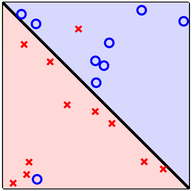
选出一个最佳 model : H_{m^*}

- goal: select $\mathcal{H}_{m^*}$ such that $g_{m^*} = \mathcal{A}_{m^*}(\mathcal{D})$ is of low $E_{\text{out}}(g_{m^*})$
- unknown E_{out} due to unknown $P(\mathbf{x})$ & $P(y|\mathbf{x})$, as always **:-)**
- arguably the **most important** practical problem of ML

如何选 model

how to select? **visually?**
—no, remember Lecture 7? :-)

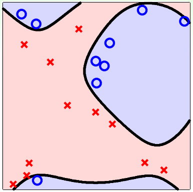
Model Selection by Best E_{in}



$\mathcal{H}_1$

select by best E_{in}?

$$m^* = \underset{1 \le m \le M}{\operatorname{argmin}} (E_m = E_{\text{in}}(\mathcal{A}_m(\mathcal{D})))$$



$\mathcal{H}_2$

选更複雜

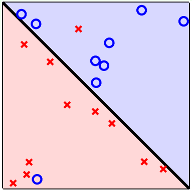
选择不做 regularization

- Φ_{1126} always more preferred over Φ_1;
  $\lambda = 0$ always more preferred over $\lambda = 0.1$—**overfitting?**
- if $\mathcal{A}_1$ minimizes E_{in} over $\mathcal{H}_1$ and $\mathcal{A}_2$ minimizes E_{in} over $\mathcal{H}_2$,
  $\Longrightarrow g_{m^*}$ achieves minimal E_{in} over $\mathcal{H}_1 \cup \mathcal{H}_2$ 等於在考慮 $H_1 \cup H_2$ 的 hypothesis
  $\Longrightarrow$ 'model selection + learning' pays $d_{\text{VC}}(\mathcal{H}_1 \cup \mathcal{H}_2)$
  —**bad generalization?**

用Ein选很危險

selecting by E_{in} is **dangerous**

Model Selection by Best E_{test}

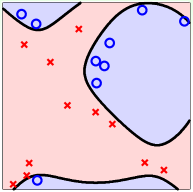


$\mathcal{H}_1$

select by best E_{test}, which is evaluated on a fresh $\mathcal{D}_{\text{test}}$?

$$m^* = \underset{1 \le m \le M}{\operatorname{argmin}} \left(E_m = E_{\text{test}}(\mathcal{A}_m(\mathcal{D})) \right)$$

选Etest最小的model



$\mathcal{H}_2$

- generalization guarantee (finite-bin Hoeffding):

$$E_{\text{out}}(g_{m^*}) \le E_{\text{test}}(g_{m^*}) + O\left(\sqrt{\frac{\log M}{N_{\text{test}}}}\right)$$

—**yes! strong guarantee :-)** Eout被bound住.

- but where is $\mathcal{D}_{\text{test}}$?—**your boss's safe, maybe? :-(**

找得到Dtest?

selecting by E_{test} is **infeasible** and **cheating**

Comparison between E_{in} and E_{test}

in-sample error E_{in}

- calculated from $\mathcal{D}$
- feasible on hand
- 'contaminated' as $\mathcal{D}$ also used by $\mathcal{A}_m$ to 'select' g_m

test error E_{test}

- calculated from $\mathcal{D}_{\text{test}}$
- infeasible in boss's safe
- 'clean' as $\mathcal{D}_{\text{test}}$ never used for selection before

something in between: E_{val}

- calculated from $\mathcal{D}_{\text{val}} \subset \mathcal{D}$ validation set
- **feasible** on hand
- 'clean' **if** $\mathcal{D}_{\text{val}}$ never used by $\mathcal{A}_m$ before

selecting by E_{val}: **legal cheating :-)**

Questions?

Validation Set $\mathcal{D}_{\text{val}}$

training data 拆成 Dtrain+Dval.

$$
\begin{array}{ccccc}
E_{\text{in}}(h) & & & & E_{\text{val}}(h) \\
\uparrow & & & & \uparrow \\
\underbrace{\mathcal{D}}_{\text{size } N} & \rightarrow & \underbrace{\mathcal{D}_{\text{train}}}_{\text{size } N-K} & \cup & \underbrace{\mathcal{D}_{\text{val}}}_{\text{size } K} \\
\downarrow & & \downarrow & & \\
g_m = \mathcal{A}_m(\mathcal{D}) & & g_m^- = \mathcal{A}_m(\mathcal{D}_{\text{train}}) & &
\end{array}
$$

valid

訓練model

data 比較小

- $\mathcal{D}_{\text{val}} \subset \mathcal{D}$: called **validation set**—'on-hand' simulation of test set
- to connect E_{val} with E_{out}:

  Dval 從D 平均隨机抽樣.

  $\mathcal{D}_{\text{val}} \overset{iid}{\sim} P(\mathbf{x}, y) \Longleftarrow$ select K examples from $\mathcal{D}$ at random
- to make sure $\mathcal{D}_{\text{val}}$ 'clean':
  feed only $\mathcal{D}_{\text{train}}$ to $\mathcal{A}_m$ for model selection

$$E_{\text{out}}(g_m^-) \le E_{\text{val}}(g_m^-) + O\left(\sqrt{\frac{\log M}{K}}\right)$$

Model Selection by Best E_{val}

选 val error 最小的 model

$$m^* = \underset{1 \le m \le M}{\operatorname{argmin}} \left(E_m = E_{\text{val}}(\mathcal{A}_m(\mathcal{D}_{\text{train}})) \right)$$

- generalization guarantee for all m:

$$E_{\text{out}}(g_m^-) \le E_{\text{val}}(g_m^-) + O\left(\sqrt{\frac{\log M}{K}}\right)$$

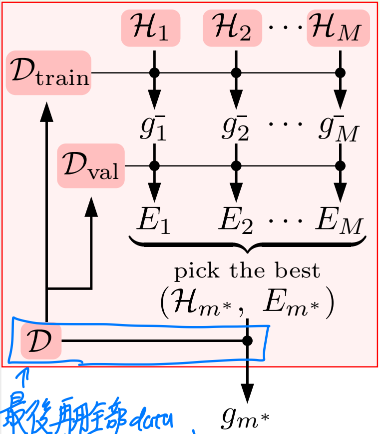
- heuristic gain from $N - K$ to N:

选完model之後,連同validation set一起下去train.

$$E_{\text{out}}\Bigg(\underbrace{g_{m^*}}_{\mathcal{A}_{m^*}(\mathcal{D})}\Bigg) \le E_{\text{out}}\Bigg(\underbrace{g_{m^*}^-}_{\mathcal{A}_{m^*}(\mathcal{D}_{\text{train}})}\Bigg)$$

data又更多了一些.

—**learning curve, remember? :-)**



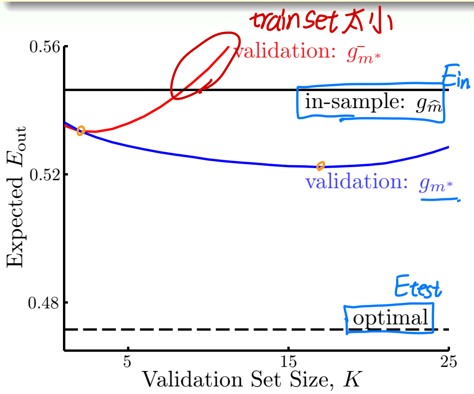
最後用全部data train.

$$E_{\text{out}}(g_{m^*}) \le E_{\text{out}}(g_{m^*}^-) \le E_{\text{val}}(g_{m^*}^-) + O\left(\sqrt{\frac{\log M}{K}}\right)$$

Validation in Practice

use validation to select between $\mathcal{H}_{\Phi_5}$ and $\mathcal{H}_{\Phi_{10}}$

5-R 10-R



$g_{m^*} < g^-_{m^*}$

- in-sample: selection with E_{in}
- optimal: cheating-selection with E_{test}
- sub-g: selection with E_{val} and report $g^-_{m^*}$
- full-g: selection with E_{val} and report g_{m^*} —$E_{out}(g_{m^*}) \le E_{out}(g^-_{m^*})$ indeed

why is sub-g worse than in-sample some time?

The Dilemma about K

validation set 大小

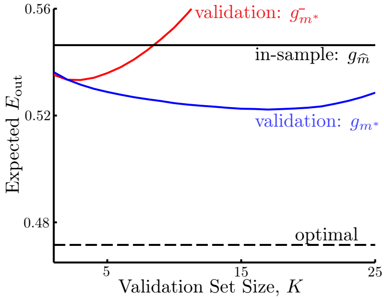
大量valid set才能保證 Eout 与 Eval 接近

reasoning of validation:

$$\underbrace{E_{\text{out}}(g) \underset{\text{(small } K)}{\approx} E_{\text{out}}(g^-)} \overbrace{\underset{\text{(large } K)}{\approx}} E_{\text{val}}(g^-)$$

但 valid set 拿太多,又会讓 training 時偏差太大

- large K: **every** $E_{\text{val}} \approx E_{\text{out}}$, but all g_m^- much worse than g_m
- small K: every $g_m^- \approx g_m$, but E_{val} far from E_{out}



20%

practical rule of thumb: $K = \frac{N}{5}$

Questions?

Extreme Case: $K = 1$ 太超小

reasoning of validation: ✓

GG!

$$E_{\text{out}}(g) \underset{\textbf{(small } K\textbf{)}}{\approx} E_{\text{out}}(g^-) \underset{\textbf{(large } K\textbf{)}}{\approx} E_{\text{val}}(g^-)$$

第N筆資料的error

- take $K = 1$? $\mathcal{D}_{\text{val}}^{(n)} = \{(\mathbf{x}_n, y_n)\}$ and $E_{\text{val}}^{(n)}(g_n^-) = \text{err}(g_n^-(\mathbf{x}_n), y_n) = e_n$
- make e_n closer to $E_{\text{out}}(g)$?—average over possible $E_{\text{val}}^{(n)}$
- leave-one-out cross validation estimate:

$$E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) = \frac{1}{N}\sum_{n=1}^{N} e_n = \underbrace{\frac{1}{N}\sum_{n=1}^{N} \text{err}(g_n^-(\mathbf{x}_n), y_n)}_{\text{平均}}$$

leave-one-out cross validation.

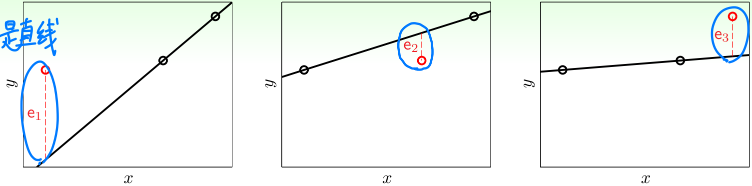
希望

hope: $E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) \approx E_{\text{out}}(g)$

假設只有3個 data point

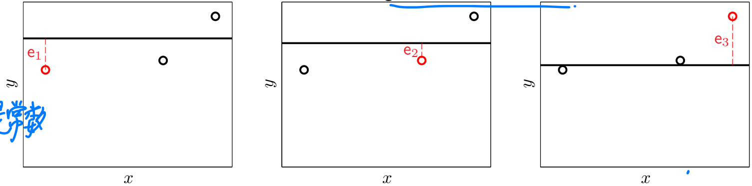
Illustration of Leave-One-Out

hypothesis是直线



$$E_{\text{loocv}}(\text{linear}) = \frac{1}{3}(e_1 + e_2 + e_3)$$

hypothesis是常數



$$E_{\text{loocv}}(\text{constant}) = \frac{1}{3}(e_1 + e_2 + e_3)$$

用常數 model 就夠了

which one would you choose?

$$m^* = \underset{1 \le m \le M}{\text{argmin}}(E_m = E_{\text{loocv}}(\mathcal{H}_m, \mathcal{A}_m))$$

Theoretical Guarantee of Leave-One-Out Estimate

does $E_{\text{loocv}}(\mathcal{H}, \mathcal{A})$ say something about $E_{\text{out}}(g)$?

yes, for average E_{out} on size-$(N-1)$ data

$$\underset{\mathcal{D}}{\mathcal{E}} E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) = \underset{\mathcal{D}}{\mathcal{E}} \frac{1}{N} \sum_{n=1}^{N} e_n = \frac{1}{N} \sum_{n=1}^{N} \underset{\mathcal{D}}{\mathcal{E}} e_n$$

期望值

leve one-out 的那筆data

$$= \frac{1}{N} \sum_{n=1}^{N} \underset{\mathcal{D}_n}{\mathcal{E}} \underset{(\mathbf{x}_n, y_n)}{\mathcal{E}} \boxed{\operatorname{err}(g_n^-(\mathbf{x}_n), y_n)}$$

$$= \frac{1}{N} \sum_{n=1}^{N} \underset{\mathcal{D}_n}{\mathcal{E}} \boxed{E_{\text{out}}(g_n^-)}$$

平均值

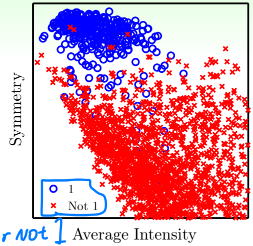
$$= \frac{1}{N} \sum_{n=1}^{N} \overline{E_{\text{out}}}(N-1) = \overline{E_{\text{out}}}(N-1)$$

$E_{\text{out}}(g^-)$

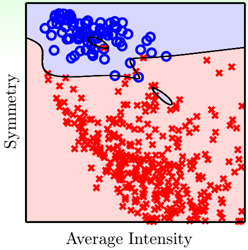
expected $E_{\text{loocv}}(\mathcal{H}, \mathcal{A})$ says something about expected $E_{\text{out}}(g^-)$

—often called 'almost unbiased estimate of $E_{\text{out}}(g)$'

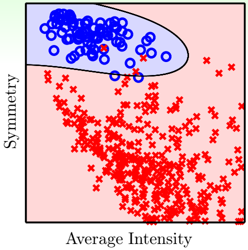
Leave-One-Out in Practice



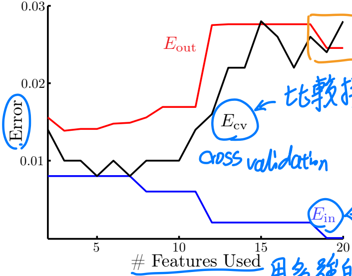
1 or not 1



select by E_{in}



select by E_{loocv}



overfit

比較接近 E_{out}

E_{loocv} much better than E_{in}

← fitting problem 當然是愈高次愈好

用多強的 feature transformation.

Questions?

Disadvantages of Leave-One-Out Estimate

Computation

$$E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) = \frac{1}{N}\sum_{n=1}^{N} e_n = \frac{1}{N}\sum_{n=1}^{N} \text{err}(g_n^-(\mathbf{x}_n), y_n)$$

- N 'additional' training per model, not always feasible in practice
- except 'special case' like analytic solution for linear regression

linear regression 適合用 LOOCV

E_{loocv}: not often used practically

不好搞

V-fold Cross Validation

how to **decrease computation need** for cross validation?

- essence of leave-one-out cross validation: partition $\mathcal{D}$ to N parts, taking $N-1$ for training and 1 for validation orderly
- V-fold cross-validation: random-partition of $\mathcal{D}$ **to V equal parts**,

$$\overbrace{\mathcal{D}_1\ \mathcal{D}_2\ \mathcal{D}_3\ \mathcal{D}_4\ \mathcal{D}_5\ \mathcal{D}_6\ \mathcal{D}_7\ \mathcal{D}_8\ \mathcal{D}_9\ \mathcal{D}_{10}}^{\mathcal{D}}$$

$$\text{train} \qquad \text{validate} \qquad \text{train}$$

只拿一份V，做validate

take $V-1$ for training and 1 for validation orderly

$$E_{\text{cv}}(\mathcal{H}, \mathcal{A}) = \frac{1}{V}\sum_{v=1}^{V} E_{\text{val}}^{(v)}(g_v^-)$$

平均

- selection by E_{cv}: $m^* = \underset{1 \le m \le M}{\text{argmin}}(E_m = E_{\text{cv}}(\mathcal{H}_m, \mathcal{A}_m))$

cross validation.

practical rule of thumb: $\boxed{V = 10}$

20%? 10%?

Final Words on Validation

'Selecting' Validation Tool

- ***V*-Fold** generally preferred over single validation if computation allows
  cross validation.
- **5-Fold or 10-Fold** generally works well: 通常不用 LOO
  not necessary to trade V-Fold with Leave-One-Out

Nature of Validation

- all training models: select among hypotheses
- all validation schemes: select among finalists
- all testing methods: just evaluate

validation still more optimistic than testing

do not fool yourself and others :-),
report test result, not best validation result

Questions?

Summary

1. When Can Machines Learn?
2. Why Can Machines Learn?
3. How Can Machines Learn?
4. How Can Machines Learn **Better**?

Lecture 8: Combatting Overfitting (2)

- Model Selection Problem
  dangerous by E_{in} and dishonest by E_{test}
- Validation
  select with $E_{\text{val}}(\mathcal{A}_m(\mathcal{D}_{\text{train}}))$ while returning $\mathcal{A}_{m^*}(\mathcal{D})$
- Leave-One-Out Cross Validation
  huge computation for almost unbiased estimate
- *V*-Fold Cross Validation
  reasonable computation and performance

- **next: something 'up my sleeve'**