Lecture 15: Validation

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 14: Regularization

minimizes augmented error, where the added regularizer effectively limits model complexity

Lecture 15: Validation

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

Model Selection Problem

So Many Models Learned

Even Just for Binary Classification . . .

\[
\mathcal{A} \in \{ \text{PLA, pocket, linear regression, logistic regression} \} \\
T \in \{ 100, 1000, 10000 \} \\
\eta \in \{ 1, 0.01, 0.0001 \} \\
\Phi \in \{ \text{linear, quadratic, poly-10, Legendre-poly-10} \} \\
\Omega(\mathbf{w}) \in \{ \text{L2 regularizer, L1 regularizer, symmetry regularizer} \} \\
\lambda \in \{ 0, 0.01, 1 \}
\]

in addition to your favorite combination, may need to try other combinations to get a good \( g \)
Model Selection Problem

- given: $M$ models $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_M$, each with corresponding algorithm $A_1, A_2, \ldots, A_M$
- goal: select $\mathcal{H}_{m^*}$ such that $g_{m^*} = A_{m^*}(D)$ is of low $E_{out}(g_{m^*})$
- unknown $E_{out}$ due to unknown $P(x) & P(y|x)$, as always :-)
- arguably the most important practical problem of ML

which one do you prefer? :-)

how to select? visually? —no, remember Lecture 12? :-)

Hsuan-Tien Lin (NTU CSIE)
Model Selection by Best $E_{\text{in}}$

select by best $E_{\text{in}}$?

$$m^* = \arg\min_{1 \leq m \leq M} (E_m = E_{\text{in}}(A_m(D)))$$

- $\Phi_{1126}$ always more preferred over $\Phi_1$;
- $\lambda = 0$ always more preferred over $\lambda = 0.1$—overfitting?

- if $A_1$ minimizes $E_{\text{in}}$ over $\mathcal{H}_1$ and $A_2$ minimizes $E_{\text{in}}$ over $\mathcal{H}_2$,  
  $\implies g_{m^*}$ achieves minimal $E_{\text{in}}$ over $\mathcal{H}_1 \cup \mathcal{H}_2$
  $\implies$ ‘model selection + learning’ pays $d_{\text{VC}}(\mathcal{H}_1 \cup \mathcal{H}_2)$—bad generalization?

selecting by $E_{\text{in}}$ is dangerous
Model Selection by Best $E_{\text{test}}$

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

$$m^* = \arg\min_{1 \leq m \leq M} (E_m = E_{\text{test}}(A_m(D)))$$

- generalization guarantee (finite-bin Hoeffding):

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

—yes! strong guarantee :-)

- but where is $D_{\text{test}}$?—your boss’s safe, maybe? :-(

selecting by $E_{\text{test}}$ is infeasible and cheating
## Validation

### Model Selection Problem

### Comparison between $E_{\text{in}}$ and $E_{\text{test}}$

<table>
<thead>
<tr>
<th>in-sample error $E_{\text{in}}$</th>
<th>test error $E_{\text{test}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>• calculated from $\mathcal{D}$</td>
<td>• calculated from $\mathcal{D}_{\text{test}}$</td>
</tr>
<tr>
<td>• feasible on hand</td>
<td>• infeasible in boss’s safe</td>
</tr>
<tr>
<td>• ‘contaminated’ as $\mathcal{D}$ also used by $\mathcal{A}_m$ to ‘select’ $g_m$</td>
<td>• ‘clean’ as $\mathcal{D}_{\text{test}}$ never used for selection before</td>
</tr>
</tbody>
</table>

### Something in between: $E_{\text{val}}$

• calculated from $\mathcal{D}_{\text{val}} \subset \mathcal{D}$
• feasible on hand
• ‘clean’ if $\mathcal{D}_{\text{val}}$ never used by $\mathcal{A}_m$ before

selecting by $E_{\text{val}}$: legal cheating :-)
For $\mathcal{X} = \mathbb{R}^d$, consider two hypothesis sets, $\mathcal{H}_+$ and $\mathcal{H}_-$. The first hypothesis set contains all perceptrons with $w_1 \geq 0$, and the second hypothesis set contains all perceptrons with $w_1 \leq 0$. Denote $g_+$ and $g_-$ as the minimum-$E_{\text{in}}$ hypothesis in each hypothesis set, respectively.

Which statement below is true?

1. If $E_{\text{in}}(g_+) < E_{\text{in}}(g_-)$, then $g_+$ is the minimum-$E_{\text{in}}$ hypothesis of all perceptrons in $\mathbb{R}^d$.
2. If $E_{\text{test}}(g_+) < E_{\text{test}}(g_-)$, then $g_+$ is the minimum-$E_{\text{test}}$ hypothesis of all perceptrons in $\mathbb{R}^d$.
3. The two hypothesis sets are disjoint.
4. None of the above
For $\mathcal{X} = \mathbb{R}^d$, consider two hypothesis sets, $\mathcal{H}_+$ and $\mathcal{H}_-$. The first hypothesis set contains all perceptrons with $w_1 \geq 0$, and the second hypothesis set contains all perceptrons with $w_1 \leq 0$. Denote $g_+$ and $g_-$ as the minimum-$E_{\text{in}}$ hypothesis in each hypothesis set, respectively. Which statement below is true?

1. If $E_{\text{in}}(g_+) < E_{\text{in}}(g_-)$, then $g_+$ is the minimum-$E_{\text{in}}$ hypothesis of all perceptrons in $\mathbb{R}^d$.

2. If $E_{\text{test}}(g_+) < E_{\text{test}}(g_-)$, then $g_+$ is the minimum-$E_{\text{test}}$ hypothesis of all perceptrons in $\mathbb{R}^d$.

3. The two hypothesis sets are disjoint.

4. None of the above

Reference Answer: 1

Note that the two hypothesis sets are not disjoint (sharing ‘$w_1 = 0$’ perceptrons) but their union is all perceptrons.
Validation Set $\mathcal{D}_{\text{val}}$

\[
\begin{align*}
E_{\text{in}}(h) & \uparrow \\
\mathcal{D} & \quad \rightarrow \\
\text{size } N & \quad \mathcal{D}_{\text{train}} \\
\downarrow & \\
g_m = \mathcal{A}_m(\mathcal{D}) & \quad g_m^- = \mathcal{A}_m(\mathcal{D}_{\text{train}}) \\
\end{align*}
\]

- $\mathcal{D}_{\text{val}} \subset \mathcal{D}$: called **validation set**—‘on-hand’ simulation of test set
- to connect $E_{\text{val}}$ with $E_{\text{out}}$: $\mathcal{D}_{\text{val}} \overset{iid}{\sim} P(x, y) \iff$ 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

\[
E_{\text{out}}(g_m^-) \leq E_{\text{val}}(g_m^-) + O\left(\sqrt{\frac{\log M}{K}}\right)
\]
Model Selection by Best $E_{\text{val}}$

$m^* = \text{argmin}_{1 \leq m \leq M} (E_m = E_{\text{val}}(A_m(D_{\text{train}})))$

- generalization guarantee for all $m$:
  \[ E_{\text{out}}(g_m^-) \leq E_{\text{val}}(g_m^-) + O\left(\sqrt{\log \frac{M}{K}}\right) \]

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

\[
E_{\text{out}}\left(\begin{array}{c}
g_{m^*}^- \\
A_{m^*}(D)
\end{array}\right) \leq E_{\text{out}}\left(\begin{array}{c}
g_{m^*}^- \\
A_{m^*}(D_{\text{train}})
\end{array}\right)
\]

—learning curve, remember? :-)

\[
E_{\text{out}}(g_{m^*}) \leq E_{\text{out}}(g_{m^*}^-) \leq E_{\text{val}}(g_{m^*}^-) + O\left(\sqrt{\frac{\log M}{K}}\right)
\]
Validation in Practice

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

- 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 $g_{m*}$

Indeed

$$E_{out}(g_{m*}) \leq E_{out}(g_{m*})$$

why is sub-$g$ worse than in-sample some time?
The Dilemma about $K$

reasoning of validation:

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

(smaller $K$) \quad (larger $K$)

- 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 $E_{\text{val}}$ far from $E_{\text{out}}$

practical rule of thumb: $K = \frac{N}{5}$
For a learning model that takes $N^2$ seconds of training when using $N$ examples, what is the total amount of seconds needed when running the whole validation procedure with $K = \frac{N}{5}$ on 25 such models with different parameters to get the final $g_{m^*}$?

1. $6N^2$
2. $17N^2$
3. $25N^2$
4. $26N^2$
For a learning model that takes $N^2$ seconds of training when using $N$ examples, what is the total amount of seconds needed when running the whole validation procedure with $K = \frac{N}{5}$ on 25 such models with different parameters to get the final $g_{m^*}$?

1. $6N^2$
2. $17N^2$
3. $25N^2$
4. $26N^2$

Reference Answer: 2

To get all the $g_m^-$, we need $\frac{16}{25}N^2 \cdot 25$ seconds. Then to get $g_{m^*}$, we need another $N^2$ seconds. So in total we need $17N^2$ seconds.
Extreme Case: $K = 1$

**Reasoning of validation:**

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

(Small $K$): (Large $K$)

- take $K = 1$? $\mathcal{D}_{\text{val}}^{(n)} = \{(x_n, y_n)\}$ and $E_{\text{val}}^{(n)}(g_n^-) = \text{err}(g_n^-(x_n), y_n) = e_n$
- make $e_n$ closer to $E_{\text{out}}(g)$?—平均 over possible $E_{\text{val}}^{(n)}$

**Leave-one-out cross validation estimate:**

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

**Hope:**

$$E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) \approx E_{\text{out}}(g)$$
Illustration of Leave-One-Out

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

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

which one would you choose?

\[ m^* = \arg\min_{1 \leq m \leq M} (E_m = E_{\text{loocv}}(H_m, A_m)) \]
Theoretical Guarantee of Leave-One-Out Estimate

\[ E_{\text{loocv}}(H, A) \] says something about \( E_{\text{out}}(g) \)?

Yes, for average \( E_{\text{out}} \) on size-\((N - 1)\) data

\[
\mathcal{E}_D E_{\text{loocv}}(H, A) = \frac{1}{N} \sum_{n=1}^{N} e_n = \frac{1}{N} \sum_{n=1}^{N} \mathcal{E}_D e_n
\]

\[
= \frac{1}{N} \sum_{n=1}^{N} \mathcal{E}_D \mathcal{E}_{D_n(x_n, y_n)} \text{err}(g_n^-(x_n), y_n)
\]

\[
= \frac{1}{N} \sum_{n=1}^{N} \mathcal{E}_D E_{\text{out}}(g_n^-)
\]

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

Expected \( E_{\text{loocv}}(H, A) \) says something about expected \( E_{\text{out}}(g^-) \)—often called ‘almost unbiased estimate of \( E_{\text{out}}(g) \)’
Leave-One-Out in Practice

- Select by $E_{\text{in}}$ & $E_{\text{loocv}}$
- $E_{\text{loocv}}$ much better than $E_{\text{in}}$
Consider three examples \((x_1, y_1), (x_2, y_2), (x_3, y_3)\) with \(y_1 = 1\), \(y_2 = 5\), \(y_3 = 7\). If we use \(E_{\text{loocv}}\) to estimate the performance of a learning algorithm that predicts with the average \(y\) value of the data set—the optimal constant prediction with respect to the squared error. What is \(E_{\text{loocv}}\) (squared error) of the algorithm?

1. 0
2. \(\frac{56}{9}\)
3. \(\frac{60}{9}\)
4. 14
Consider three examples \((x_1, y_1), (x_2, y_2), (x_3, y_3)\) with \(y_1 = 1, y_2 = 5, y_3 = 7\). If we use \(E_{\text{loocv}}\) to estimate the performance of a learning algorithm that predicts with the average \(y\) value of the data set—the optimal constant prediction with respect to the squared error. What is \(E_{\text{loocv}}\) (squared error) of the algorithm?

- 0
- \(\frac{56}{9}\)
- \(\frac{60}{9}\)
- 14

Reference Answer: 4

This is based on a simple calculation of 
\[e_1 = (1 - 6)^2, e_2 = (5 - 4)^2, e_3 = (7 - 3)^2.\]
Validation

V-Fold Cross Validation

Disadvantages of Leave-One-Out Estimate

**Computation**

\[
E_{\text{loocv}}(\mathcal{H}, \mathcal{A}) = \frac{1}{N} \sum_{n=1}^{N} e_n = \frac{1}{N} \sum_{n=1}^{N} \text{err}(g_n^-(x_n), y_n)
\]

- \(N\) ‘additional’ training per model, not always feasible in practice
- except ‘special case’ like analytic solution for linear regression

**Stability**—due to variance of single-point estimates

\[E_{\text{loocv}}: \text{not often used practically}\]
**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,

  \[
  \mathcal{D} = \{D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8, D_9, D_{10}\}
  \]

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

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

- selection by $E_{cv}$: $m^* = \arg\min_{1 \leq m \leq M} (E_m = E_{cv}(\mathcal{H}_m, \mathcal{A}_m))$

practical rule of thumb: $V = 10$
Final Words on Validation

‘Selecting’ Validation Tool

- **V-Fold** generally preferred over single validation if computation allows
- **5-Fold or 10-Fold** generally works well: 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
For a learning model that takes $N^2$ seconds of training when using $N$ examples, what is the total amount of seconds needed when running 10-fold cross validation on 25 such models with different parameters to get the final $g_{m^*}$?

1. $\frac{47}{2} N^2$
2. $47 N^2$
3. $\frac{407}{2} N^2$
4. $407 N^2$
For a learning model that takes $N^2$ seconds of training when using $N$ examples, what is the total amount of seconds needed when running 10-fold cross validation on 25 such models with different parameters to get the final $g_m^*$?

1. $\frac{47}{2}N^2$
2. $47N^2$
3. $\frac{407}{2}N^2$
4. $407N^2$

Reference Answer: 3

To get all the $E_{cv}$, we need $\frac{81}{100}N^2 \cdot 10 \cdot 25$ seconds. Then to get $g_m^*$, we need another $N^2$ seconds. So in total we need $\frac{407}{2}N^2$ seconds.
Summary

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

Lecture 14: Regularization

Lecture 15: Validation

- Model Selection Problem
dangerous by $E_{\text{in}}$ and dishonest by $E_{\text{test}}$

- Validation
select with $E_{\text{val}}(A_m(D_{\text{train}}))$ while returning $A_{m^*}(D)$

- Leave-One-Out Cross Validation
huge computation for almost unbiased estimate

- V-Fold Cross Validation
reasonable computation and performance

• next: something ‘up my sleeve’