

CSE4088 Introduction to Machine Learning

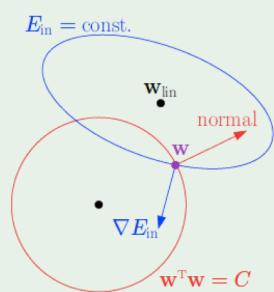
Validation

Slides are adopted from lecture notes of Yaser Abu-Mostafa

Review of last lecture

- Regularization

constrained \longrightarrow unconstrained



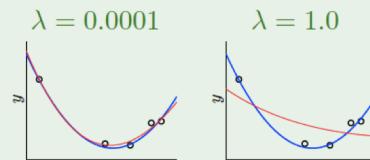
$$\text{Minimize } E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N} \mathbf{w}^T \mathbf{w}$$

- Choosing a regularizer

$$E_{\text{aug}}(h) = E_{\text{in}}(h) + \frac{\lambda}{N} \Omega(h)$$

$\Omega(h)$: heuristic \rightarrow smooth, simple h
most used: **weight decay**

λ : principled; validation



Outline

- The validation set and the difference from test set
- Model selection is the main task of validation.
- Cross validation

Validation versus regularization

In one form or another, $E_{\text{out}}(h) = E_{\text{in}}(h) + \text{overfit penalty}$

Regularization:

$$E_{\text{out}}(h) = E_{\text{in}}(h) + \underbrace{\text{overfit penalty}}_{\text{regularization estimates this quantity}}$$

Validation:

$$\underbrace{E_{\text{out}}(h)}_{\text{validation estimates this quantity}} = E_{\text{in}}(h) + \text{overfit penalty}$$

Analyzing the estimate of E_{out}

On out-of-sample point (\mathbf{x}, y) , the error is $\mathbf{e}(h(\mathbf{x}), y)$

$$\text{Squared error: } (h(\mathbf{x}) - y)^2$$

$$\text{Binary error: } \llbracket h(\mathbf{x}) \neq y \rrbracket$$

$$\mathbb{E} [\mathbf{e}(h(\mathbf{x}), y)] = E_{\text{out}}(h)$$

$$\text{var} [\mathbf{e}(h(\mathbf{x}), y)] = \sigma^2$$

From a point to a set

On a validation set $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_K, y_K)$, the error is $E_{\text{val}}(h) = \frac{1}{K} \sum_{k=1}^K \mathbf{e}(h(\mathbf{x}_k), y_k)$

$$\mathbb{E} [E_{\text{val}}(h)] = \frac{1}{K} \sum_{k=1}^K \mathbb{E} [\mathbf{e}(h(\mathbf{x}_k), y_k)] = E_{\text{out}}(h)$$

$$\text{var} [E_{\text{val}}(h)] = \frac{1}{K^2} \sum_{k=1}^K \text{var} [\mathbf{e}(h(\mathbf{x}_k), y_k)] = \frac{\sigma^2}{K}$$

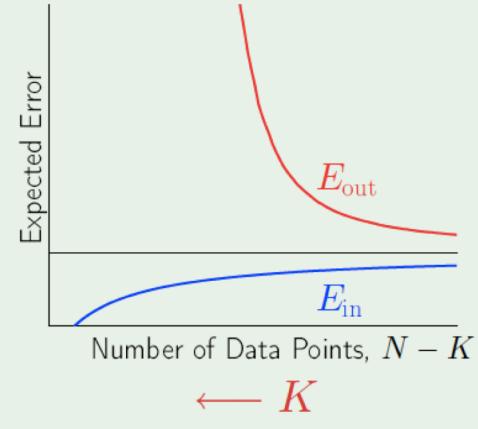
$$E_{\text{val}}(h) = E_{\text{out}}(h) \pm O\left(\frac{1}{\sqrt{K}}\right)$$

K is taken out of N

Given the data set $\mathcal{D} = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$

$\underbrace{K \text{ points}}_{\mathcal{D}_{\text{val}}} \rightarrow \text{validation} \quad \underbrace{N - K \text{ points}}_{\mathcal{D}_{\text{train}}} \rightarrow \text{training}$

$O\left(\frac{1}{\sqrt{K}}\right)$: Small $K \implies$ bad estimate
Large $K \implies ?$



K is put back into N

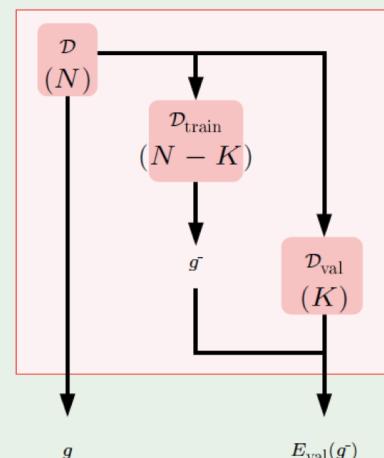
$$\begin{array}{ccc} \mathcal{D} & \longrightarrow & \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}} \\ \downarrow & \downarrow & \downarrow \\ N & & N - K & K \end{array}$$

$$\mathcal{D} \implies g \quad \mathcal{D}_{\text{train}} \implies g^-$$

$$E_{\text{val}} = E_{\text{val}}(g^-) \quad \text{Large } K \implies \text{bad estimate!}$$

Rule of Thumb:

$$K = \frac{N}{5}$$



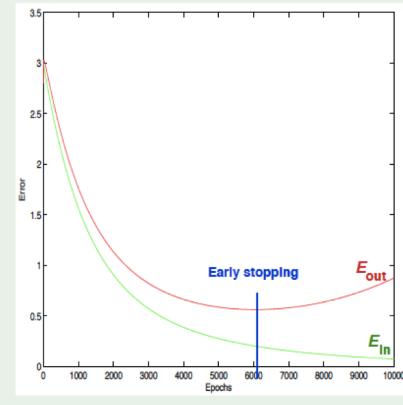
Why «validation»

\mathcal{D}_{val} is used to make learning choices

If an estimate of E_{out} affects learning:

the set is no longer a **test** set!

It becomes a **validation** set



What's the difference?

Test set is unbiased; validation set has optimistic bias

Two hypotheses h_1 and h_2 with $E_{\text{out}}(h_1) = E_{\text{out}}(h_2) = 0.5$

Error estimates e_1 and e_2 uniform on $[0, 1]$

Pick $h \in \{h_1, h_2\}$ with $e = \min(e_1, e_2)$

$E(e) < 0.5$ optimistic bias

For a proof see:

<https://danieltakeshi.github.io/2016/09/25/the-expectation-of-the-minimum-of-iid-uniform-random-variables/>

Outline

- The validation set
- Model selection
- Cross validation

Using \mathcal{D}_{val} more than once

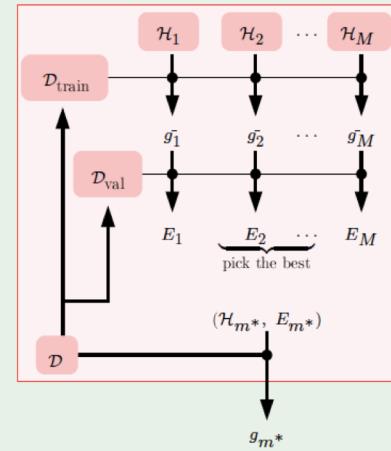
M models $\mathcal{H}_1, \dots, \mathcal{H}_M$

Use $\mathcal{D}_{\text{train}}$ to learn \bar{g}_m for each model

Evaluate \bar{g}_m using \mathcal{D}_{val} :

$$E_m = E_{\text{val}}(\bar{g}_m); \quad m = 1, \dots, M$$

Pick model $m = m^*$ with smallest E_m

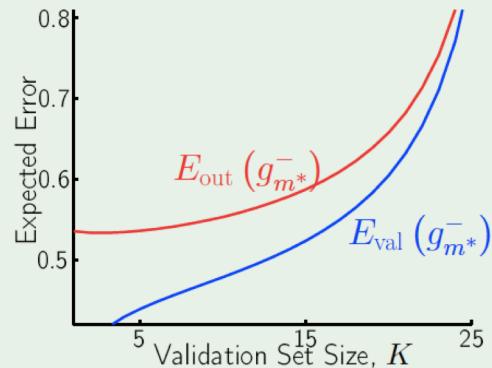


The bias

We selected the model \mathcal{H}_{m^*} using \mathcal{D}_{val}

$E_{\text{val}}(g_{m^*}^-)$ is a biased estimate of $E_{\text{out}}(g_{m^*}^-)$

Illustration: selecting between 2 models



How much bias

For M models: $\mathcal{H}_1, \dots, \mathcal{H}_M$ \mathcal{D}_{val} is used for “training” on the **finalists model**:

$$\mathcal{H}_{\text{val}} = \{g_1^-, g_2^-, \dots, g_M^-\}$$

Back to Hoeffding and VC!

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

regularization λ early-stopping T

Rule of thumb: If your validation set consists of 100 samples, you can choose 2-3 parameters. If you choose 20 parameters, you are contaminating.

Data contamination

Error estimates: E_{in} , E_{test} , E_{val}

Contamination: Optimistic (deceptive) bias in estimating E_{out}

Training set: totally contaminated

You can not rely on E_{in} to estimate E_{out}

Validation set: slightly contaminated

Test set: totally 'clean'

If you are using data for training (or to make choices), you are contaminating its ability to estimate the real performance E_{out} .

Outline

- The validation set
- Model selection
- Cross validation

The dilemma about K

The following chain of reasoning:

$$\begin{array}{c} E_{\text{out}}(g) \approx E_{\text{out}}(g^-) \approx E_{\text{val}}(g^-) \\ (\text{small } K) \qquad (\text{large } K) \end{array}$$

highlights the dilemma in selecting K :

Can we have K both small and large? ☺

Leave one out

$N - 1$ points for training, and 1 point for validation!

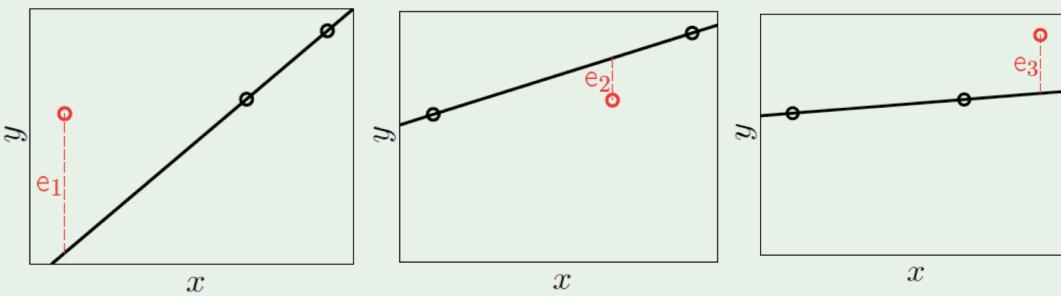
$$\mathcal{D}_n = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{n-1}, y_{n-1}), (\cancel{\mathbf{x}_n}, \cancel{y_n}), (\mathbf{x}_{n+1}, y_{n+1}), \dots, (\mathbf{x}_N, y_N)$$

Final hypothesis learned from \mathcal{D}_n is g_n^-

$$\mathbf{e}_n = E_{\text{val}}(g_n^-) = \mathbf{e}(g_n^-(\mathbf{x}_n), y_n)$$

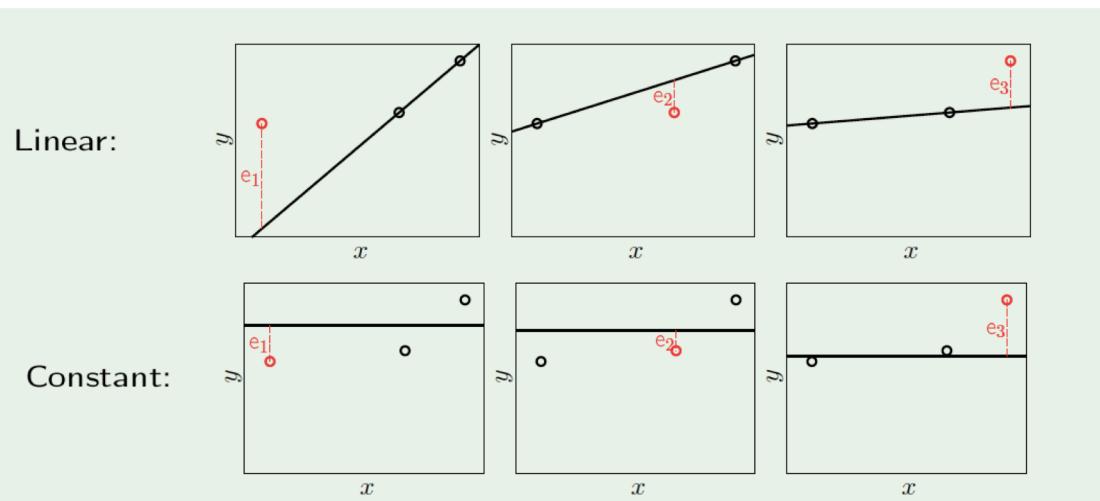
cross validation error: $E_{\text{cv}} = \frac{1}{N} \sum_{n=1}^N \mathbf{e}_n$

Illustration of cross validation

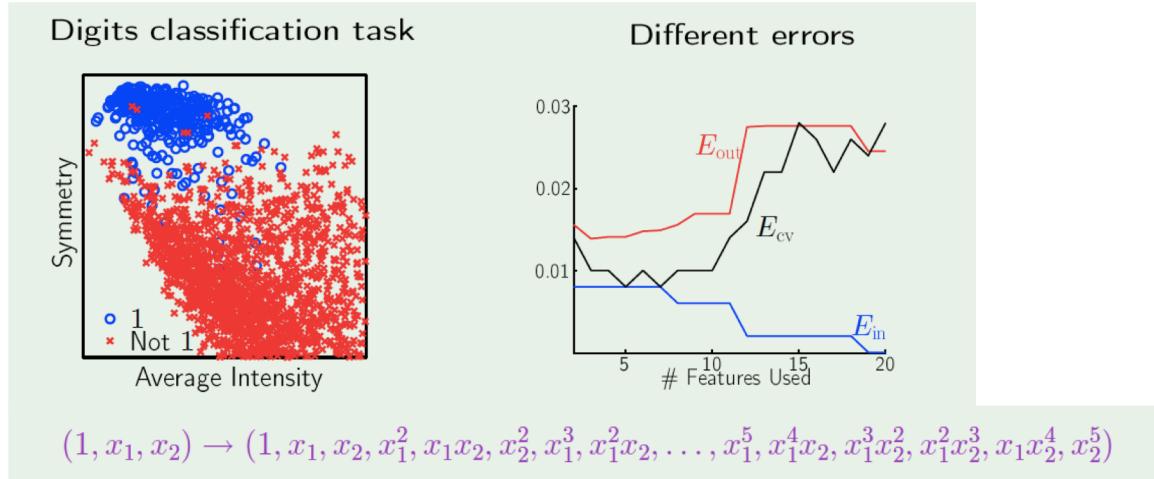


$$E_{cv} = \frac{1}{3} (e_1 + e_2 + e_3)$$

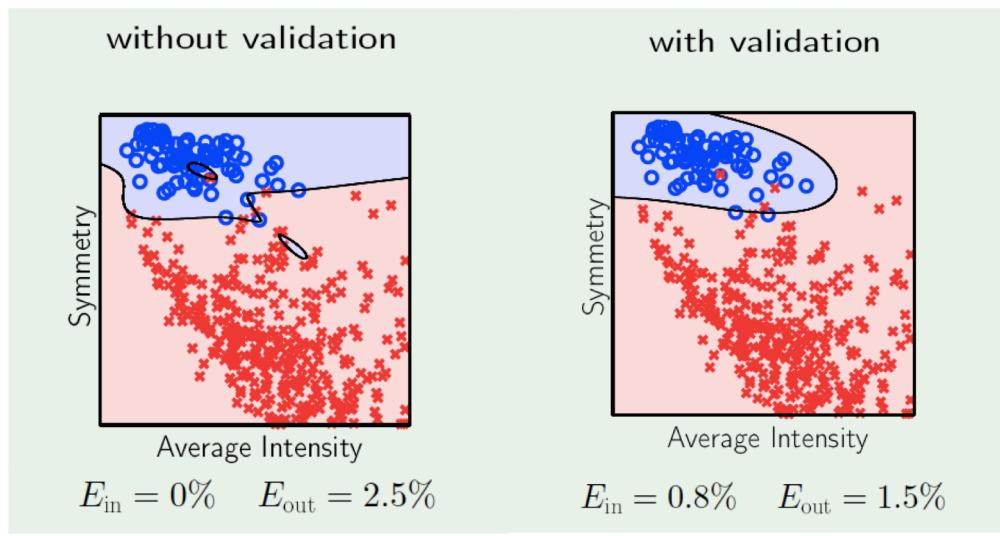
Model selection using CV



Cross validation in action



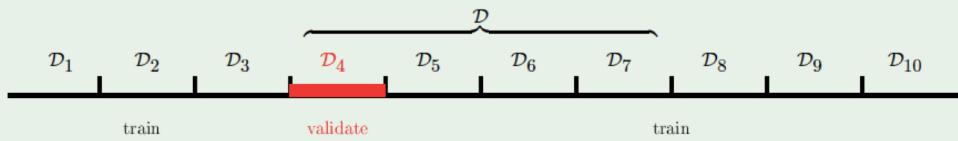
The result



Leave more than one out

Leave one out: N training sessions on $N - 1$ points each

More points for validation?



$\frac{N}{K}$ training sessions on $N - K$ points each

10-fold cross validation: $K = \frac{N}{10}$