CSE 417T: Introduction to Machine Learning

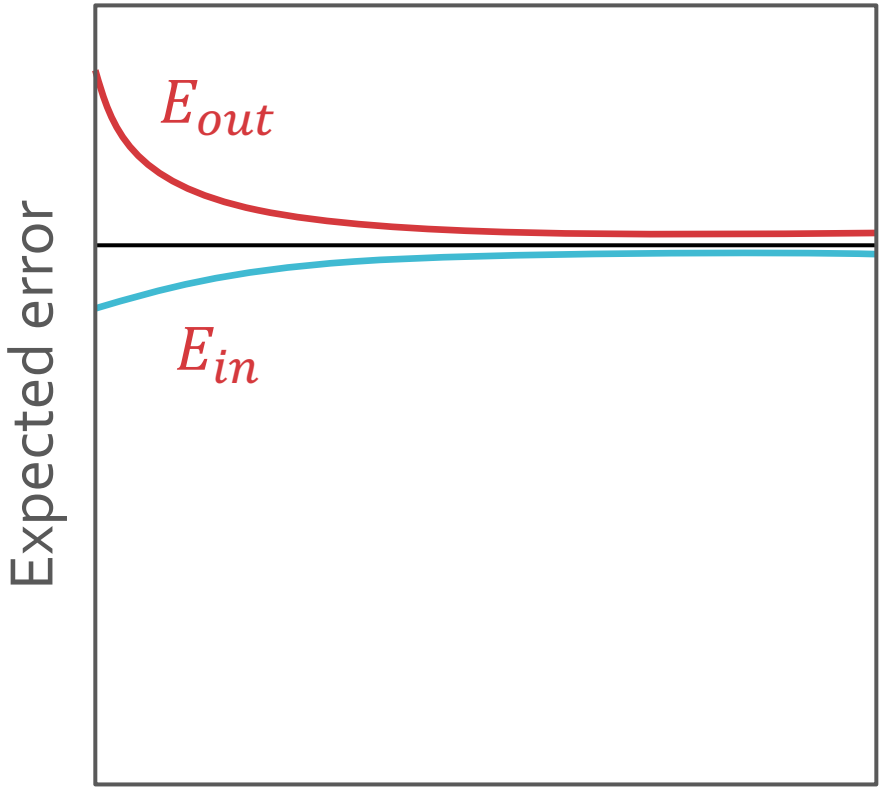
Lecture 13: Validation

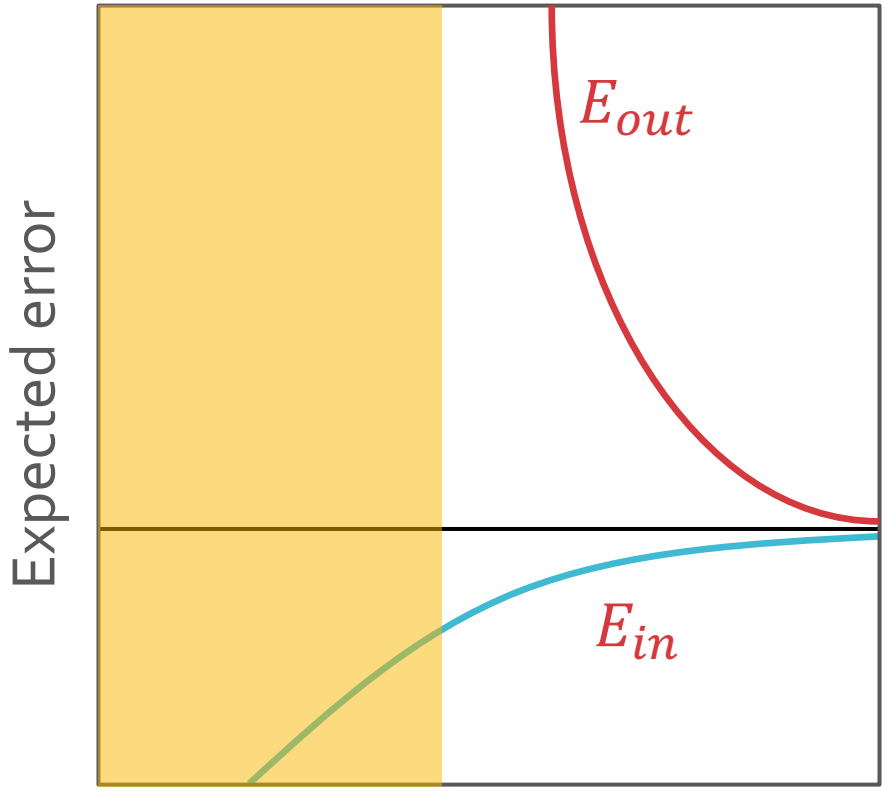
Henry Chai

10/11/18

Overfitting

- Fitting the training data “more than is warranted”
- Fitting noise rather than signal





Simple model

Complex model

Regularization

- Constrain hypothesis sets to prevent them from being able to fit noise
- Learning algorithms are optimization problems and regularization imposes constraints on that optimization

Soft Constraints: Solving for $\vec{w}_{reg}$

given $\mathcal{D} = \{(\vec{x_1}, y_1), \ldots, (\vec{x_n}, y_n)\} = (X, \vec{y})$

minimize $E_{in}(\vec{w}) = \frac{1}{n}(X\vec{w} - \vec{y})^T(X\vec{w} - \vec{y})$

subject to $\vec{w}^T\vec{w} \leq C$

$\Updownarrow$

minimize $E_{aug}(\vec{w}, \lambda_C) = E_{in}(\vec{w}) + \frac{\lambda_C}{n}\Omega(\vec{w})$

$$= E_{in}(\vec{w}) + \frac{\lambda_C}{n}\vec{w}^T\vec{w}$$

$$= \frac{(X\vec{w} - \vec{y})^T(X\vec{w} - \vec{y}) + \lambda_C\vec{w}^T\vec{w}}{n}$$

Ridge regression

$$\text{minimize } E_{aug}(\vec{w}, \lambda_C) = \frac{(X\vec{w} - \vec{y})^T(X\vec{w} - \vec{y}) + \lambda_C \vec{w}^T\vec{w}}{n}$$

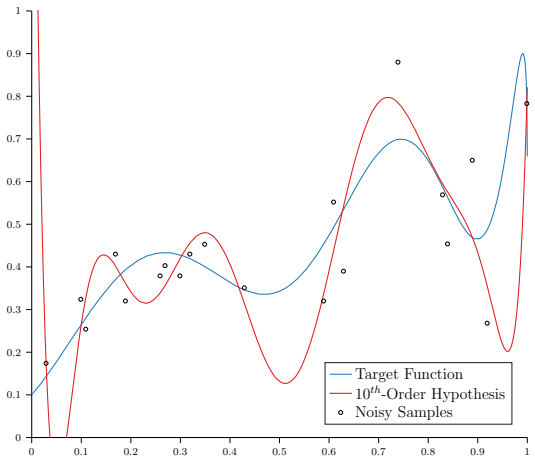
$$\nabla_{\vec{w}} E_{aug}(\vec{w}, \lambda_C) = \frac{2}{n}(X^T X\vec{w} - X^T\vec{y} + \lambda_C\vec{w})$$

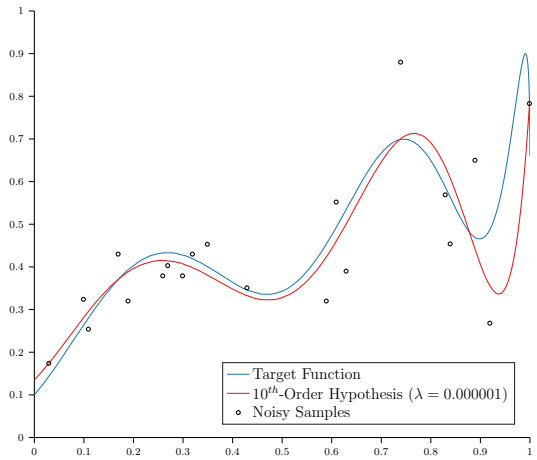
$$\nabla_{\vec{w}} E_{aug}(\vec{w}_{reg}, \lambda_C) = \frac{2}{n}(X^T X\vec{w}_{reg} - X^T\vec{y} + \lambda_C\vec{w}_{reg}) = 0$$

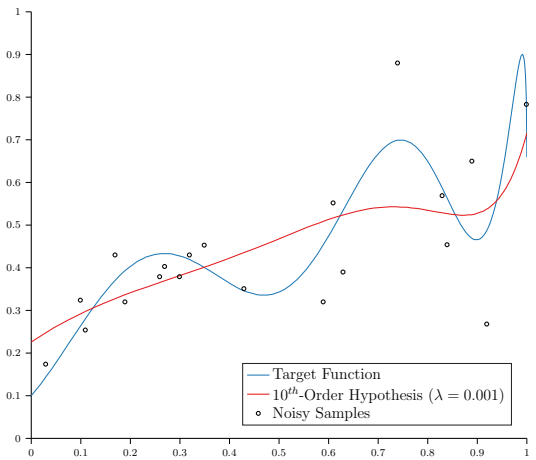
$$(X^T X + \lambda_C I_{d+1})\vec{w}_{reg} = X^T\vec{y}$$

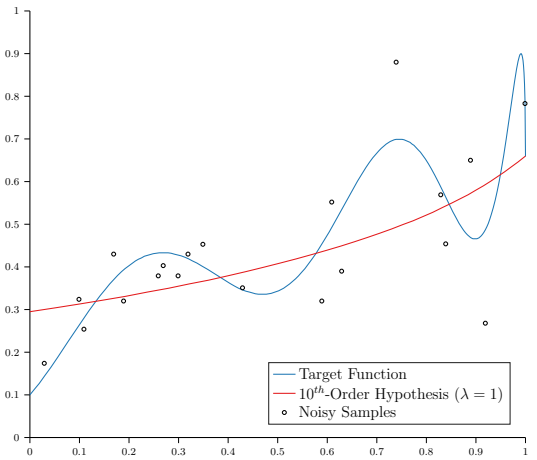
(I_{d+1} = (d+1)–by–(d+1)identity matrix)

$$\vec{w}_{reg} = (X^T X + \lambda_C I_{d+1})^{-1} X^T\vec{y}$$





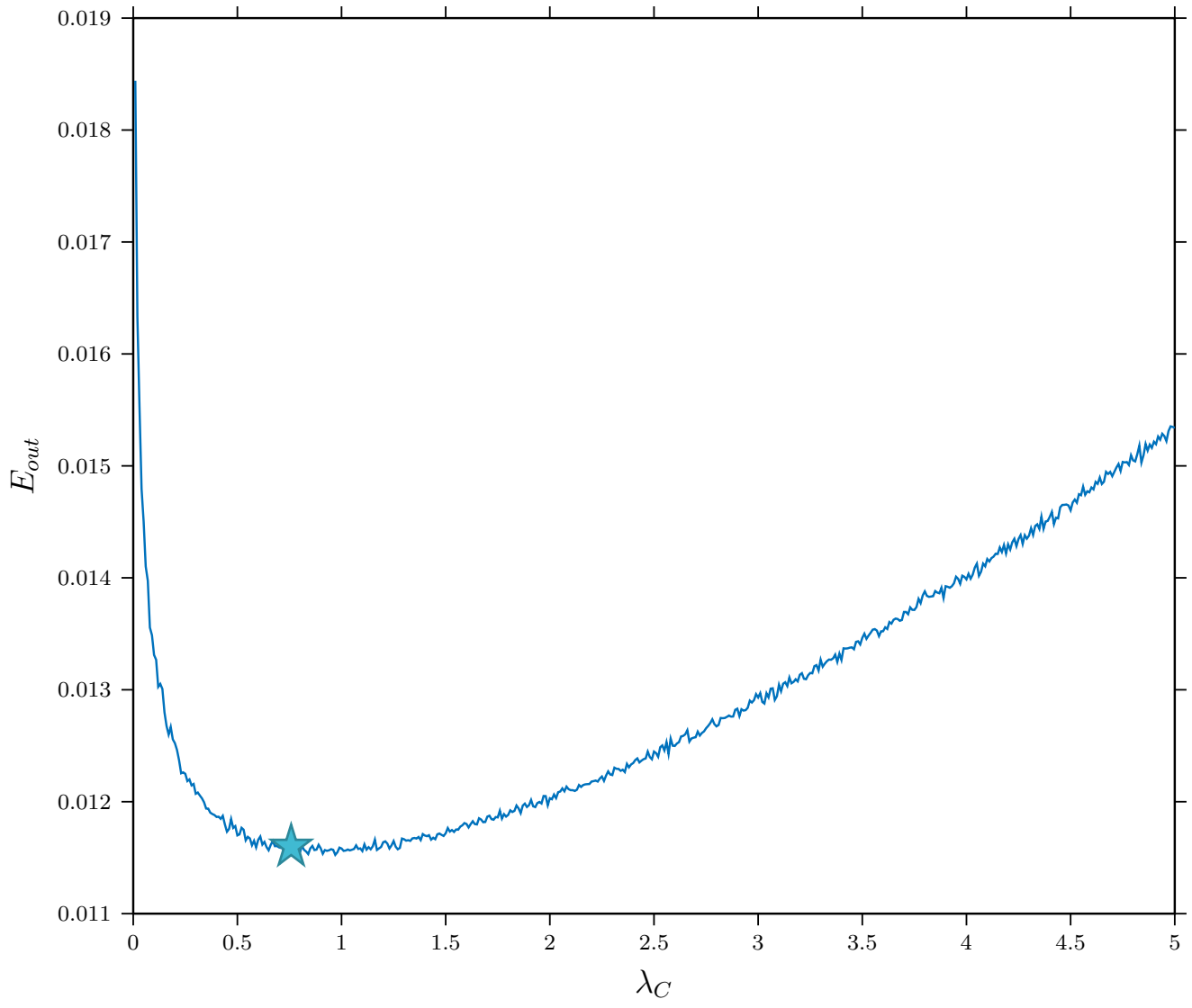


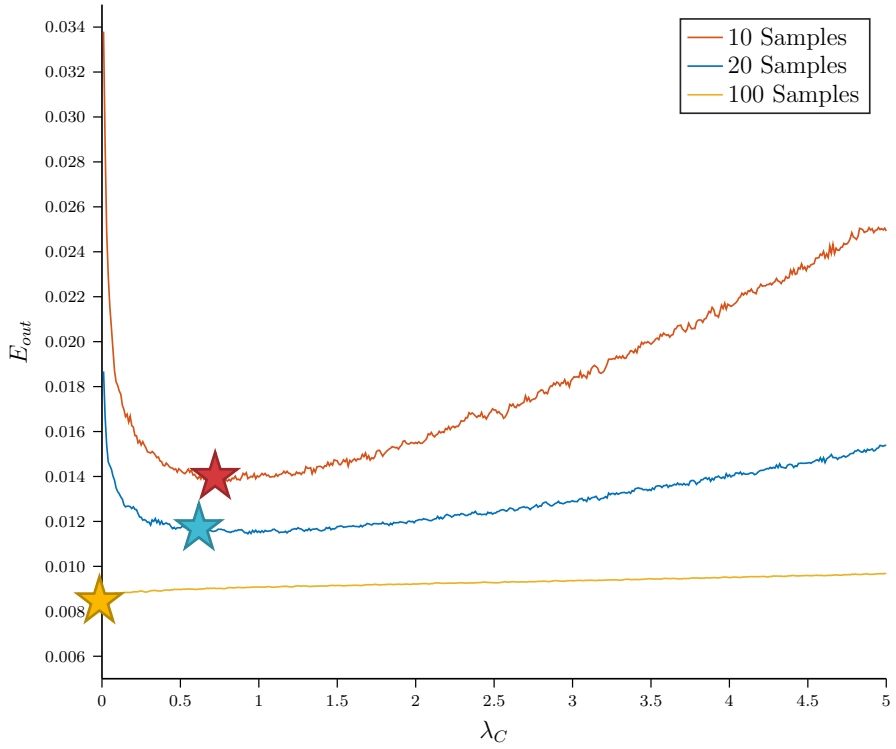


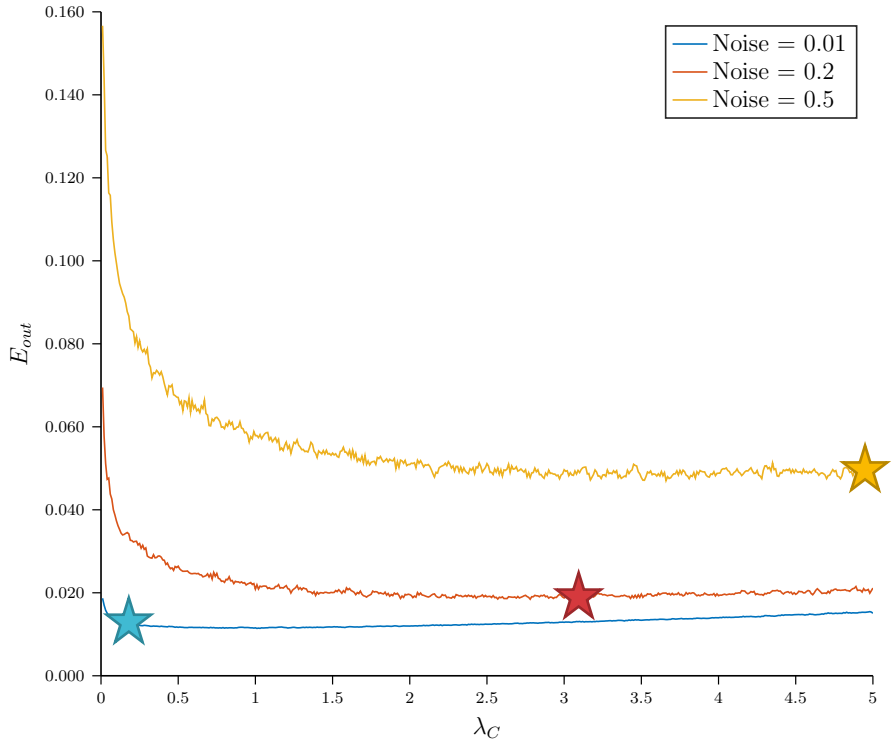
Ridge Regression: Example

$\lambda_C = 0$	$\lambda_C = 0.000001$	$\lambda_C = 0.001$	$\lambda_C = 1$
0.059	0.006	0.008	0.011
Overfit	Nice!	Wait…	Underfit

Underfitting



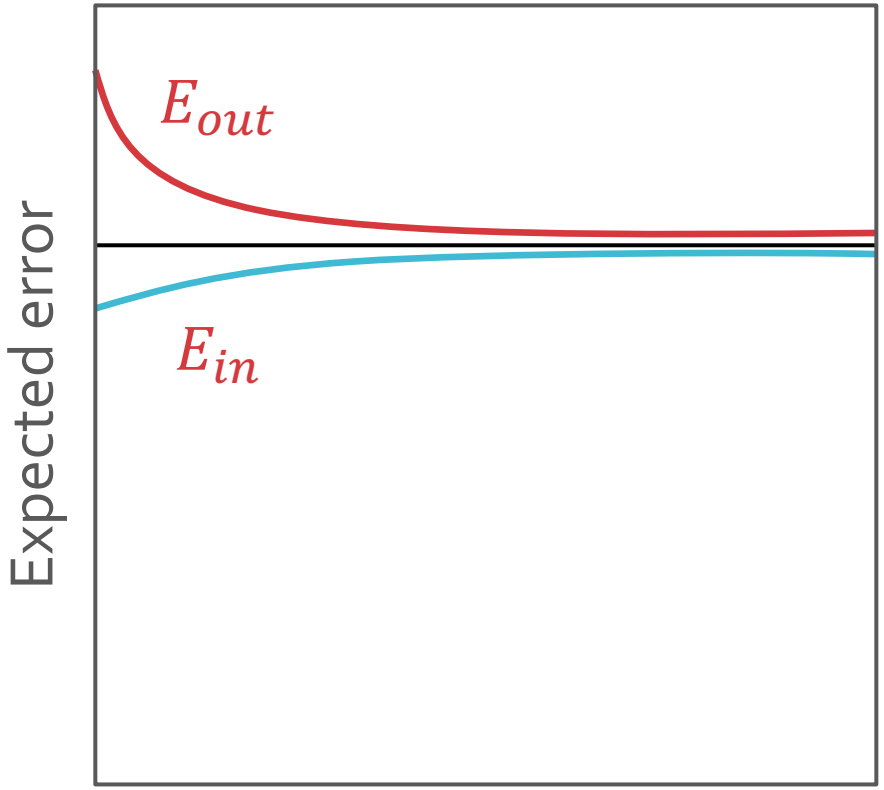


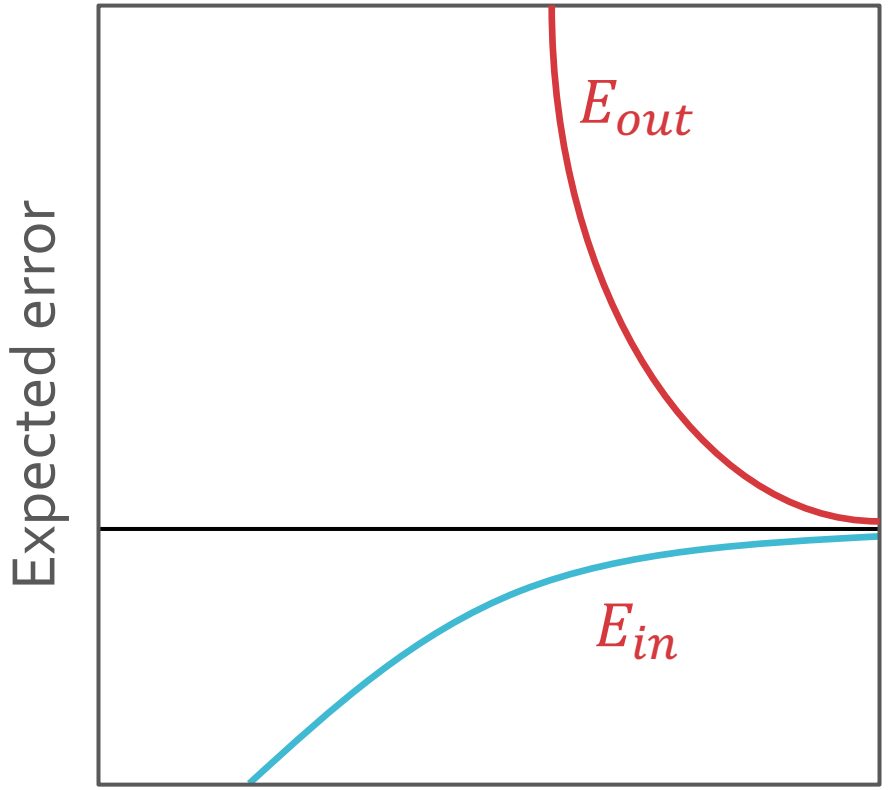


Picking λ_C

Picking Ω

- Hard: comparable to picking "the right" $\mathcal{H}$
- Generally want to constrain hypotheses to be simpler and smoother
- Even a bad regularizer can be salvaged by choosing the right λ_C





Simple model

Complex model

Estimating E_{out} instead of E_{in}

$$E_{out}(h) = E_{in}(h) + \underbrace{\text{overfit penalty}}$$

regularization estimates this quantity

$$\left(E_{aug}(\vec{w}, \lambda_C) = E_{in}(\vec{w}) + \frac{\lambda_C}{n}\Omega(\vec{w}) \right)$$

Estimating E_{out} instead of E_{in}

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

Test sets

- Estimate $E_{out}(g)$ using the error on some test dataset$\mathcal{D}_{test}$, $E_{test}(g)$
- If $\mathcal{D}_{test}$ is not involved in the training process, then $P\{|E_{test}(g) - E_{out}(g)| > \epsilon\} \leq 2e^{-2\epsilon^2 k}$ $(k = |\mathcal{D}_{test}|)$

E_{test}

given $\mathcal{D}_{test} = \{(\overrightarrow{x_1}, y_1), \dots, (\overrightarrow{x_k}, y_k)\}$

$$\mathbb{E}_{\mathcal{D}_{test}}[E_{test}(h)] = \mathbb{E}_{\mathcal{D}_{test}}\left[\frac{1}{k}\sum_{i=1}^{k} e(h(\overrightarrow{x_i}), y_i)\right]$$

$$= \frac{1}{k}\sum_{i=1}^{k} \mathbb{E}_{\mathcal{D}_{test}}[e(h(\overrightarrow{x_i}), y_i)]$$

$$= \frac{1}{k}\sum_{i=1}^{k} \mathbb{E}_{\overrightarrow{x_i}}[e(h(\overrightarrow{x_i}), y_i)] = \frac{1}{k}\sum_{i=1}^{k} E_{out}(h)$$

$$= E_{out}(h)$$

E_{test}

- $P\{|E_{test}(g) - E_{out}(g)| > \epsilon\} \leq 2e^{-2\epsilon^2 k}$
- Or…
- $E_{test}(g) - \sqrt{\frac{1}{2k}\log\left(\frac{2}{\delta}\right)} \leq E_{out}(g) \leq E_{test}(g) + \sqrt{\frac{1}{2k}\log\left(\frac{2}{\delta}\right)}$

  with probability at least $1 - \delta$

E_{test}

- $P\{|E_{test}(g) - E_{out}(g)| > \epsilon\} \leq 2e^{-2\epsilon^2 k}$
- Or…
- $E_{test}(g) - O\left(\frac{1}{\sqrt{k}}\right) \leq E_{out}(g) \leq E_{test}(g) + O\left(\frac{1}{\sqrt{k}}\right)$

  with high probability

Using test sets

- $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$ where $|\mathcal{D}_{test}| = k, |\mathcal{D}_{train}| = n - k$
- Learn some hypothesis g^- using only $\mathcal{D}_{train}$
- Estimate $E_{out}(g^-)$ using $\mathcal{D}_{test}$
- Let g be the hypothesis that would be learned using $\mathcal{D}$

Picking k

- More test data leads to a tighter bound on $E_{out}(g^-)$ but fewer training data **generally** means the learned g^- is worse i.e. $E_{out}(g^-)$ tends to increases as $n - k$ decreases
- $E_{out}(g) \leq E_{out}(g^-) \leq E_{test}(g^-) + O\left(\frac{1}{\sqrt{k}}\right)$ (with high probability)
- Return g but bound $E_{out}(g)$ using $E_{test}(g^-) + O\left(\frac{1}{\sqrt{k}}\right)$
- Practical rule of thumb: $k = \frac{n}{5}$

Model Selection

- Given $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$, suppose we have multiple candidate hypothesis sets: $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_m$
- Learn a hypothesis from each hypothesis set using only $\mathcal{D}_{train}$: $g_1^- \in \mathcal{H}_1, g_2^- \in \mathcal{H}_2, \dots, g_m^- \in \mathcal{H}_m$
- Estimate $E_{out}(g_1^-), E_{out}(g_2^-), \dots, E_{out}(g_m^-)$ using $\mathcal{D}_{test}$. Let $g_{m^*}^-$ be the hypothesis with the lowest test error: $E_{test}(g_{m^*}^-) = \min\{E_{test}(g_1^-), E_{test}(g_2^-), \dots, E_{test}(g_m^-)\}$
- Learn a hypothesis from $\mathcal{H}_{m^*}$ using $\mathcal{D}$

Test sets

- Estimate $E_{out}(g)$ using the error on some test dataset $\mathcal{D}_{test}$, $E_{test}(g)$

- If $\mathcal{D}_{test}$ is **not involved in the training process**, then $P\{|E_{test}(g) - E_{out}(g)| > \epsilon\} \leq 2e^{-2\epsilon^2 k}$ $(k = |\mathcal{D}_{test}|)$

Model Selection

- Given $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$, suppose we have multiple candidate hypothesis sets: $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_m$
- Learn a hypothesis from each hypothesis set using only $\mathcal{D}_{train}$: $g_1^- \in \mathcal{H}_1, g_2^- \in \mathcal{H}_2, \dots, g_m^- \in \mathcal{H}_m$
- Estimate $E_{out}(g_1^-), E_{out}(g_2^-), \dots, E_{out}(g_m^-)$ using $\mathcal{D}_{test}$. Let $g_{m^*}^-$ be the hypothesis with the lowest validation error: $E_{val}(g_{m^*}^-) = \min\{E_{val}(g_1^-), E_{val}(g_2^-), \dots, E_{val}(g_m^-)\}$
- Learn a hypothesis from $\mathcal{H}_{m^*}$ using $\mathcal{D}$

Validation set

- $\mathcal{D}_{train}$ is used to build a finite set of candidate hypotheses: $\mathcal{H}_{val} = \{g_1^-, g_2^-, \dots, g_m^-\}$
- $\mathcal{D}_{val}$ is used to select the hypothesis from $\mathcal{H}_{val}$: $g_{m^*}^-$
- $P\left\{\left|E_{val}(g_{m^*}^-) - E_{out}(g_{m^*}^-)\right| > \epsilon\right\} \leq 2(m)e^{-2\epsilon^2 k}$
- $E_{val}(g_{m^*}^-) - O\left(\frac{\ln(m)}{\sqrt{k}}\right) \leq E_{out}(g_{m^*}^-) \leq E_{val}(g_{m^*}^-) + O\left(\frac{\ln(m)}{\sqrt{k}}\right)$

  with high probability

E_{in} vs. E_{val} vs. E_{test}

	Bias	Relationship to E_{out}
E_{in}	Incredibly biased	VC-bound
E_{val}	Slightly biased	Hoeffding's bound (multiple hypotheses)
E_{test}	Not biased	Hoeffding's bound (single hypothesis)

Picking λ_C (for regularization)

- Given $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{val}$ and $\mathcal{H}$, suppose we multiple candidate regularization parameters: $\lambda_{C_1}, \lambda_{C_2}, \dots, \lambda_{C_m}$
- Using only $\mathcal{D}_{train}$, learn m hypotheses: $g_1^- = \text{argmin}\, E_{aug}(h, \lambda_{C_1}), g_2^- = \text{argmin}\, E_{aug}(h, \lambda_{C_2}), \dots, g_m^- = \text{argmin}\, E_{aug}(h, \lambda_{C_m})$
- Estimate $E_{out}(g_1^-), E_{out}(g_2^-), \dots, E_{out}(g_m^-)$ using $\mathcal{D}_{val}$. Let $g_{m^*}^-$ be the hypothesis with the lowest validation error: $E_{val}(g_{m^*}^-) = \min\{E_{val}(g_1^-), E_{val}(g_2^-), \dots, E_{val}(g_m^-)\}$
- Using $\mathcal{D}$, learn a hypothesis by minimizing $E_{aug}(h, \lambda_{C_{m^*}})$

Picking k

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

Picking k

True for small k

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

True for large k

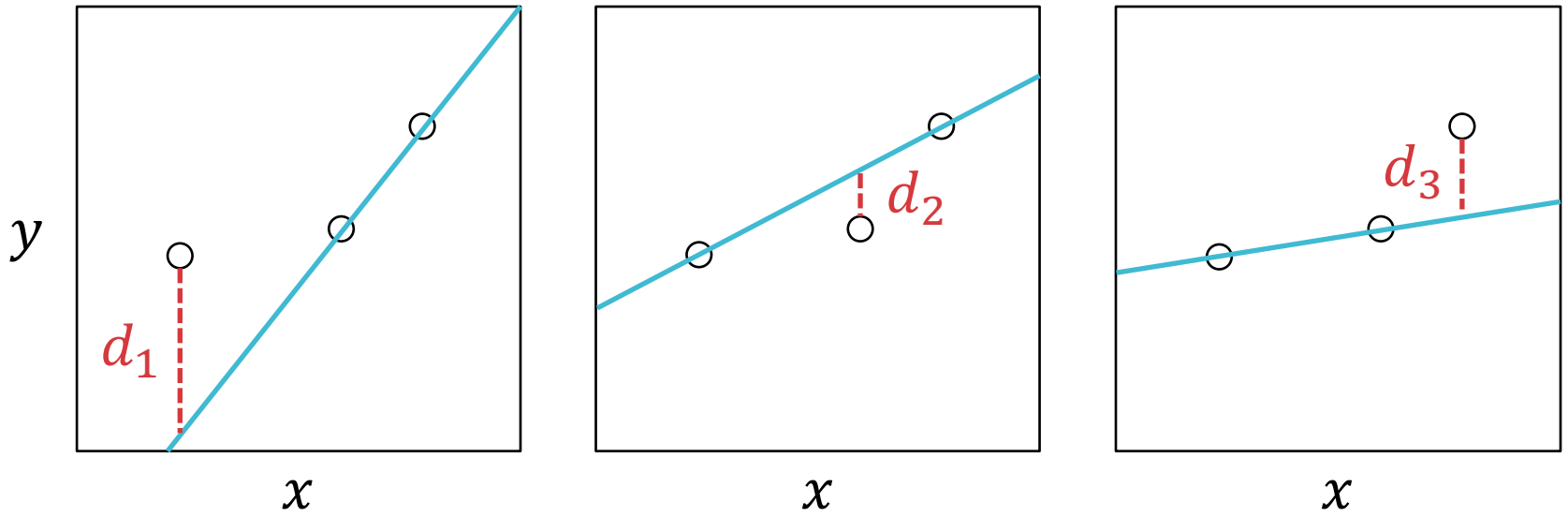
Leave-one-out cross validation

- Given $\mathcal{D} = \{(\overrightarrow{x_1}, y_1), \ldots, (\overrightarrow{x_n}, y_n)\}$, let $\mathcal{D}_{train}^{i} = \mathcal{D} \backslash (\overrightarrow{x_i}, y_i)$ (all observations except $(\overrightarrow{x_i}, y_i)$) and let $\mathcal{D}_{val}^{i} = (\overrightarrow{x_i}, y_i)$
- Let g_i^- be the hypothesis learned using only $\mathcal{D}_i$ and let $e_i = E_{val}(g_i^-) = e(g_i^-(\overrightarrow{x_i}), y_i)$
- The cross validation error is $E_{cv} = \frac{1}{n} \sum_{i=1}^{n} e_i$
- E_{cv} is almost an unbiased estimator of $E_{out}(g)$

Leave-one-out cross validation (LOOCV)

- Given $\mathcal{D} = \{(\overrightarrow{x_1}, y_1), \ldots, (\overrightarrow{x_n}, y_n)\}$, let $\mathcal{D}_{train}^{i} = \mathcal{D} \backslash (\overrightarrow{x_i}, y_i)$ (all observations except $(\overrightarrow{x_i}, y_i)$) and let $\mathcal{D}_{val}^{i} = (\overrightarrow{x_i}, y_i)$
- Let g_i^- be the hypothesis learned using only $\mathcal{D}_i$ and let $e_i = E_{val}(g_i^-) = e(g_i^-(\overrightarrow{x_i}), y_i)$
- The cross validation error is $E_{cv} = \frac{1}{n}\sum_{i=1}^{n} e_i$
- E_{cv} is an unbiased estimator of $E_{out}^{(n-1)}$

LOOCV: Example



$$E_{cv} = \frac{1}{3}(d_1^2 + d_2^2 + d_3^2)$$

Model Selection

- Given $\mathcal{D}$, suppose we have multiple candidate hypothesis sets: $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_m$
- For each hypothesis set, compute E_{cv}
- Let $g^-_{m^*}$ be the hypothesis with the lowest LOOCV error → learn a hypothesis from $\mathcal{H}_{m^*}$ using $\mathcal{D}$

Model Selection

- Given $\mathcal{D}$, suppose we have multiple candidate hypothesis sets: $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_m$
- For each hypothesis set, compute E_{cv}
- Watch all 3 Lord of the Rings extended version movies
- Let $g^-_{m^*}$ be the hypothesis with the lowest LOOCV error → learn a hypothesis from $\mathcal{H}_{m^*}$ using $\mathcal{D}$

K-fold cross validation

- Split $\mathcal{D}$ into K equally sized data sets: $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K$
- Let g_i^- be the hypothesis learned using all data sets except $\mathcal{D}_i$ and let $e_i = E_{val}(g_i^-)$ where the validation uses data set $\mathcal{D}_i$
- The K-fold cross validation error is $E_{cv_K} = \frac{1}{K}\sum_{i=1}^{K} e_i$
- Choosing between m candidate models only requires training mK times
- Practical rule of thumb: $K = 10$