# Validation and Model Selection

# A new look at $E_{\text{out}}$

**Validation is a new cure for overfitting**

$$E_{out} = E_{in} + \underbrace{\text{overfit penalty}}_{\substack{\text{VC bounds this using a complexity error bar for } \mathcal{H} \\ \text{Regularization estimates this through a heuristic complexity penalty for } g}}$$

Validation is a direct estimation to $E_{\text{out}}$

$$\underbrace{E_{out}}_{\text{validation estimates this directly}} = E_{in} + \text{overfit penalty}$$

This is a **similar estimation to given by the Test Set**

# $E_{\text{test}}$ is an unbiased estimate of $E_{\text{out}}$

$$\begin{tikzcd}
\begin{array}{c}\mathcal{D}\\ (N \text{ data points})\end{array} \arrow[d] & & \begin{array}{c}\mathcal{D}_{\text{test}}\\ (K \text{ test points})\end{array} \arrow[d] \\
g \arrow[d] \arrow[rr, "\mathrm{e}_k = \mathrm{e}(g(\mathbf{x}_k), y_k)"] & & \mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_K \arrow[d] \\
g \arrow[d] & & E_{\text{test}} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{e}_k \\
E_{\text{out}}(g) & &
\end{tikzcd}$$

$\mathrm{e}_1, \ldots, \mathrm{e}_K$ are *independent*

$$\begin{aligned}
\mathrm{Var}[E_{\text{test}}] &= \frac{1}{K^2}\sum_{k=1}^{K} \mathrm{Var}[\mathrm{e}_k] \\
&= \frac{1}{K}\mathrm{Var}[e]
\end{aligned}$$

decreases like $\frac{1}{K}$

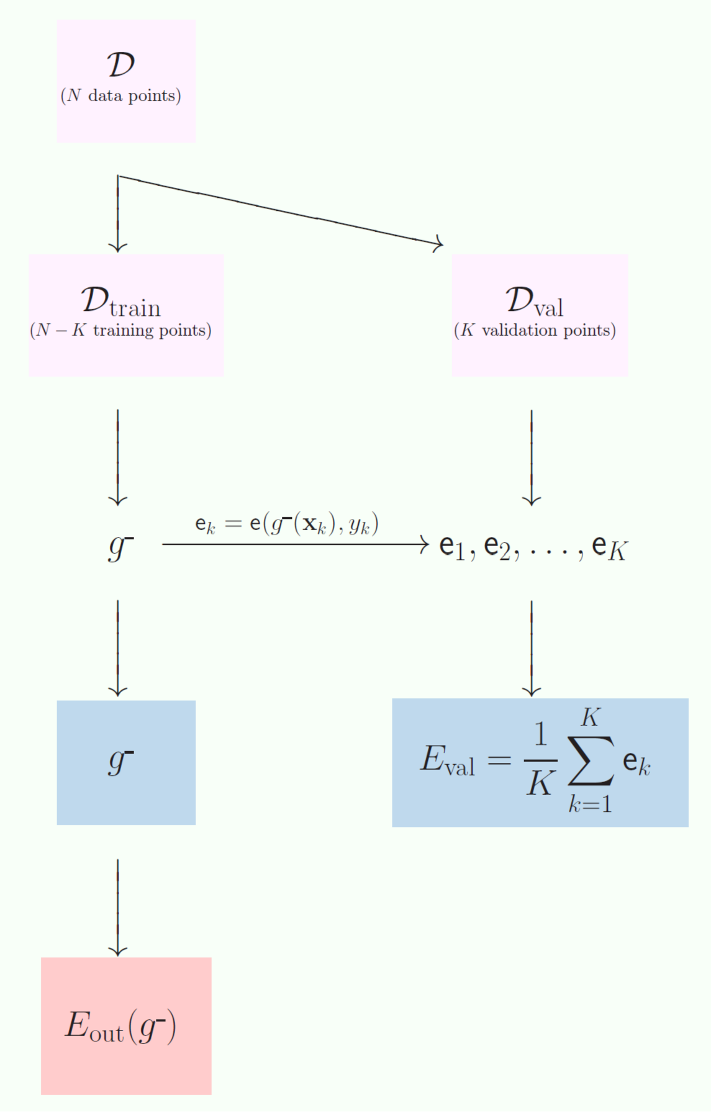
bigger $K \implies$ more reliable $E_{\text{test}}$.

**$E_{\text{test}}$ is an estimate for $E_{\text{out}}(g)$**

$$\mathbb{E}_{\mathcal{D}_{\text{test}}}[\mathrm{e}_k] = E_{\text{out}}(g)$$

$$\begin{aligned}
\mathbb{E}[E_{\text{test}}] &= \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}[\mathrm{e}_k] \\
&= \frac{1}{K}\sum_{k=1}^{K} E_{\text{out}}(g) = E_{\text{out}}(g)
\end{aligned}$$

# The Validation Set


$\mathcal{D}$
($N$ data points)
$\mathcal{D}_{\text{train}}$
($N-K$ training points)
$\mathcal{D}_{\text{val}}$
($K$ validation points)
$g^-$
$\mathrm{e}_k = \mathrm{e}(g^-(\mathbf{x}_k), y_k)$
$\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_K$
$g^-$
$E_{\text{val}} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{e}_k$
$E_{\text{out}}(g^-)$

1. Remove $K$ points from $\mathcal{D}$

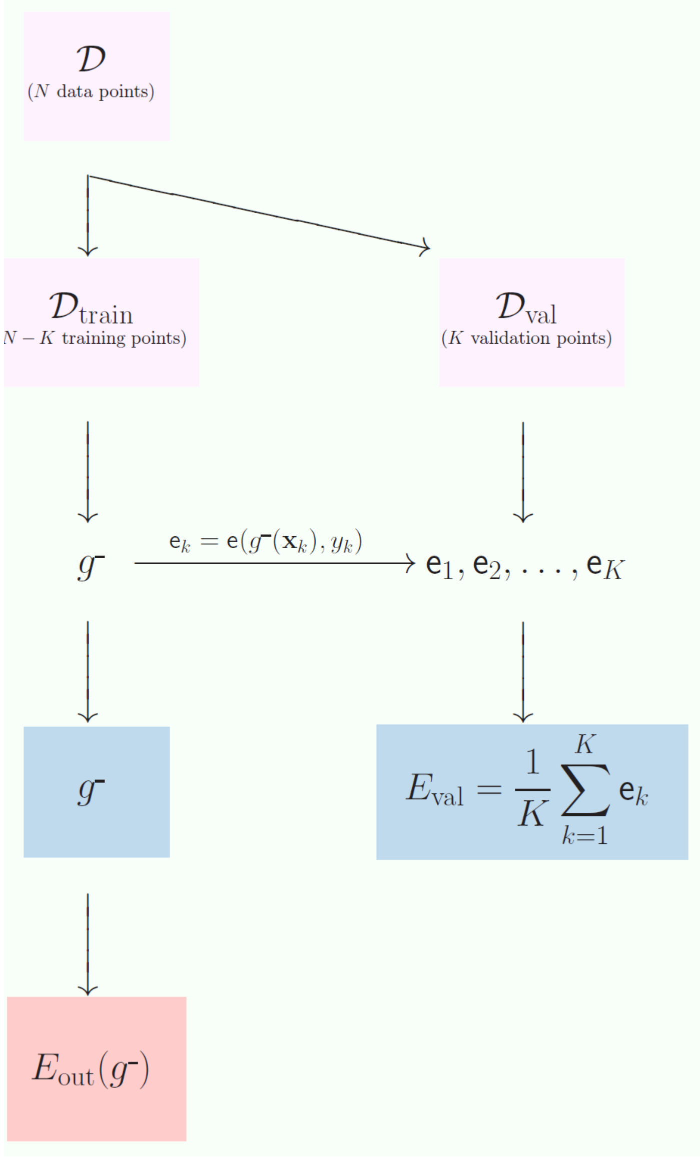
$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}}.$$

2. Learn using $\mathcal{D}_{\text{train}} \longrightarrow g^-$.

3. Test $g^-$ on $\mathcal{D}_{\text{val}} \longrightarrow E_{\text{val}}$.

4. Use error $E_{\text{val}}$ to estimate $E_{\text{out}}(g^-)$.

# The Validation Set



$\mathcal{D}$

($N$ data points)

$\mathcal{D}_{\text{train}}$

$N-K$ training points)

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

($K$ validation points)

$g^{-}$

$\mathsf{e}_k = \mathsf{e}(g^{-}(\mathbf{x}_k), y_k)$

$\mathsf{e}_1, \mathsf{e}_2, \ldots, \mathsf{e}_K$

$g^{-}$

$$E_{\text{val}} = \frac{1}{K}\sum_{k=1}^{K} \mathsf{e}_k$$

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

**$E_{\text{val}}$ is an estimate for $E_{\text{out}}(g^{-})$**

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}_{\text{val}}}[\mathsf{e}_k] &= E_{\text{out}}(g^{-})\\
\mathbb{E}[E_{\text{test}}] &= \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[\mathsf{e}_k]\\
&= \frac{1}{K}\sum_{k=1}^{K}E_{\text{out}}(g^{-}) = E_{\text{out}}(g^{-})
\end{aligned}$$

$\mathsf{e}_1, \ldots, \mathsf{e}_K$ are *independent*

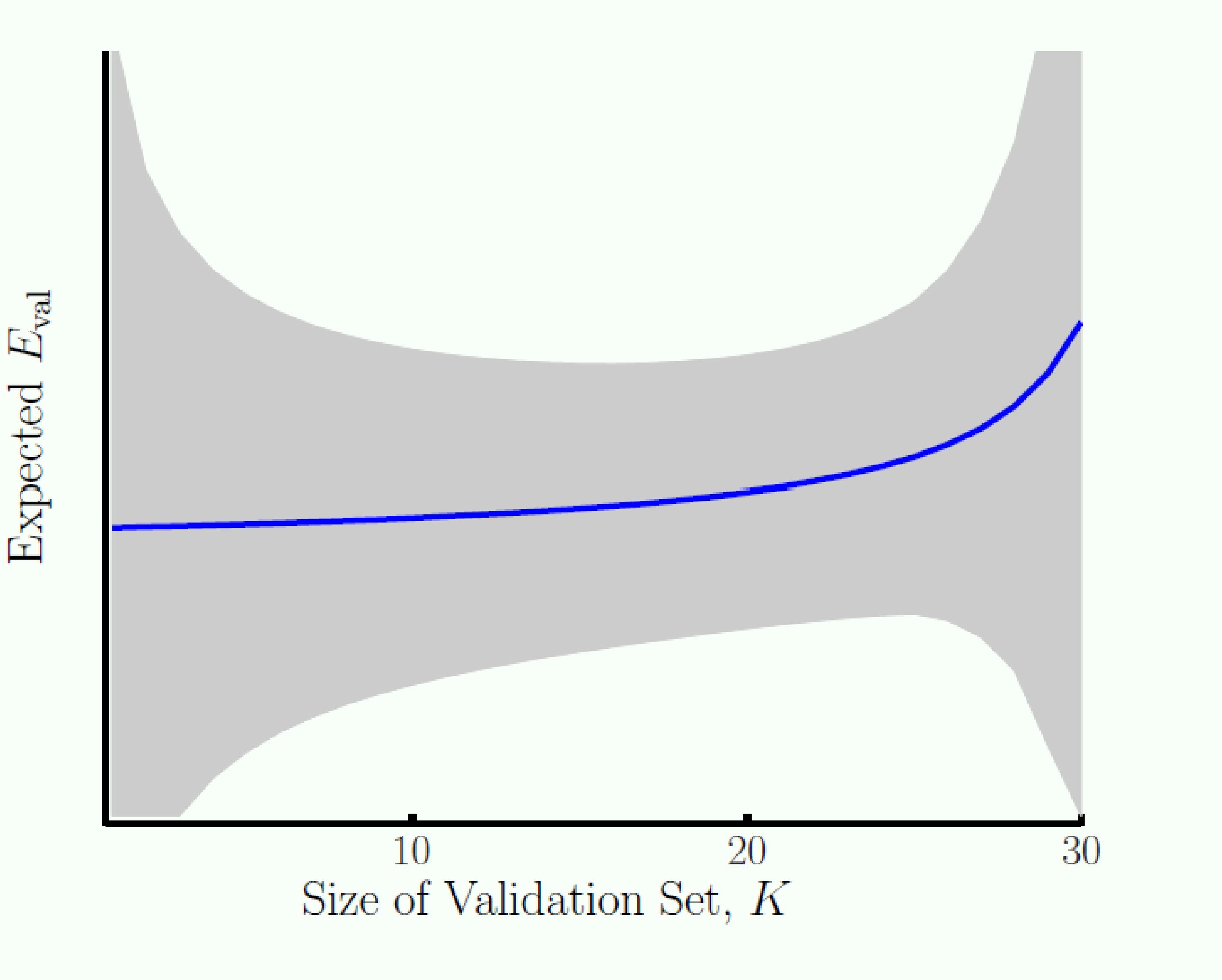
$$\begin{aligned}
\text{Var}[E_{\text{val}}] &= \frac{1}{K^2}\sum_{k=1}^{K}\text{Var}[\mathsf{e}_k]\\
&= \frac{1}{K}\text{Var}[e(g^{-})]
\end{aligned}$$

↖ decreases like $\frac{1}{K}$

depends on $g^{-}$, not $\mathcal{H}$

bigger $K \implies$ more reliable $E_{\text{val}}$?

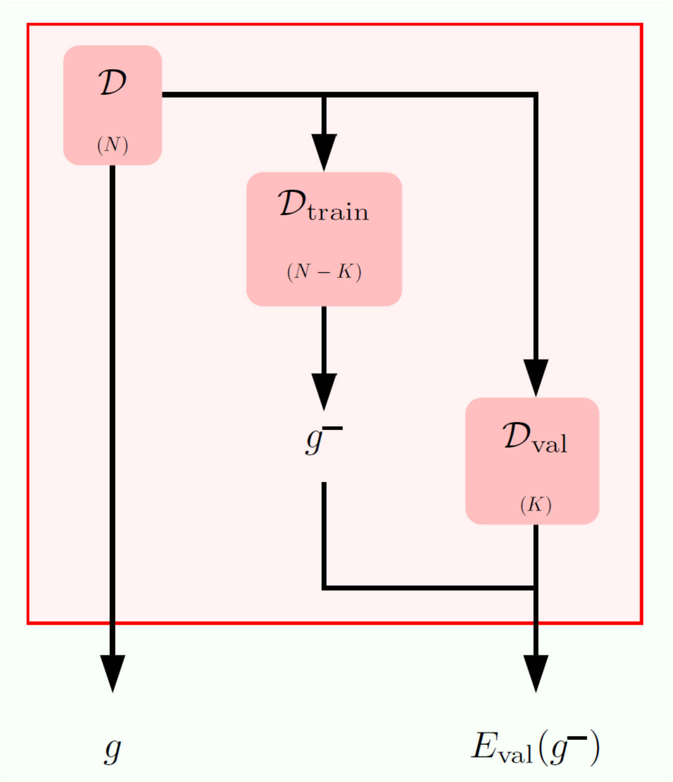
# Choosing K



This picture, associated to a simple model, shows the price to pay to put out K points to estimate $E_{val}$

Rule of thumb: $K^* = \dfrac{N}{5}$

# How to relate the K-value to our goals ?



Primary goal: output the best hypothesis trained on all data.

Secondary goal: estimate $E_{\text{out}}(g)$ using $E_{val}(g^-)$

Let's focus on the Secondary Goal:

Hoeffding bound

$$E_{out}(g) \leq E_{out}(g^-) \leq E_{val}(g^-) + \mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$$

Learning curves

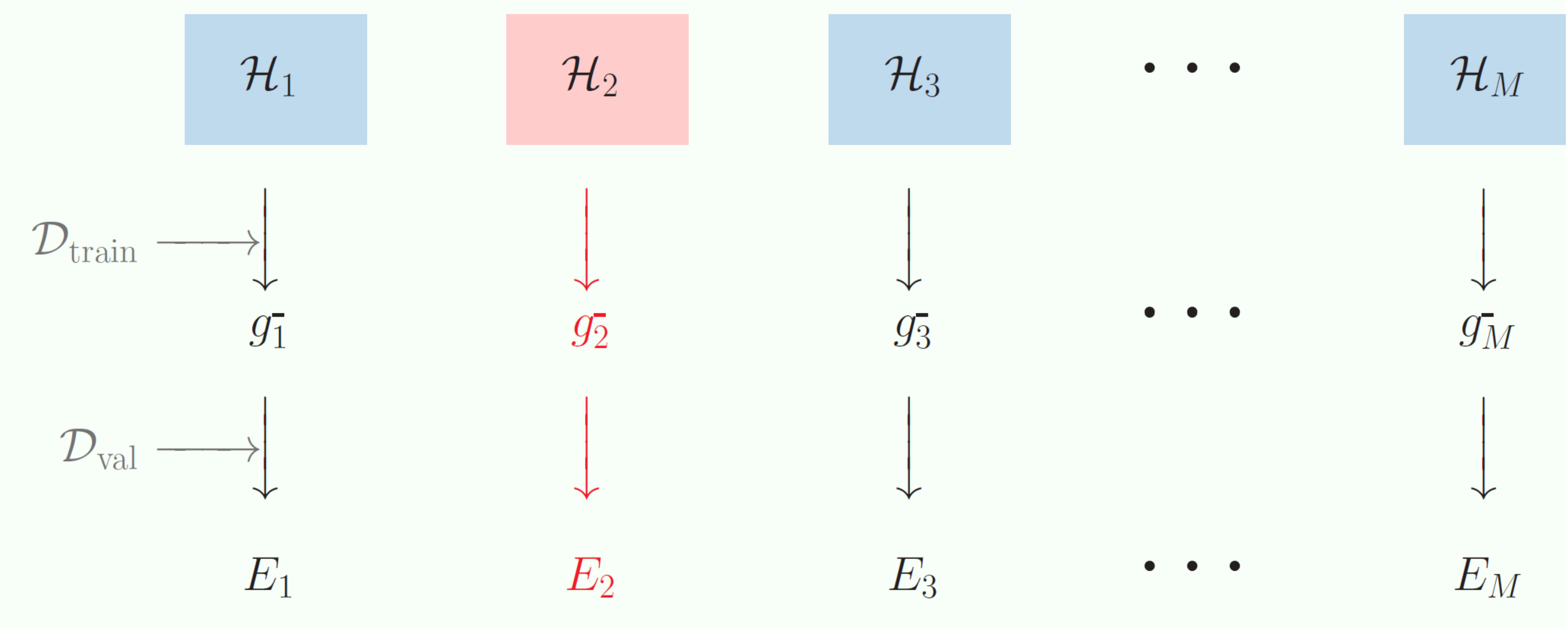
Primary goal: Once we have a good estimation for $E_{out}$ all samples can be used for training.

Clearly the estimation for $E_{out}$ will be pessimistic respect to performance of our best hypothesis., but that is a good result

# Model Selection

## The most important use of Validation

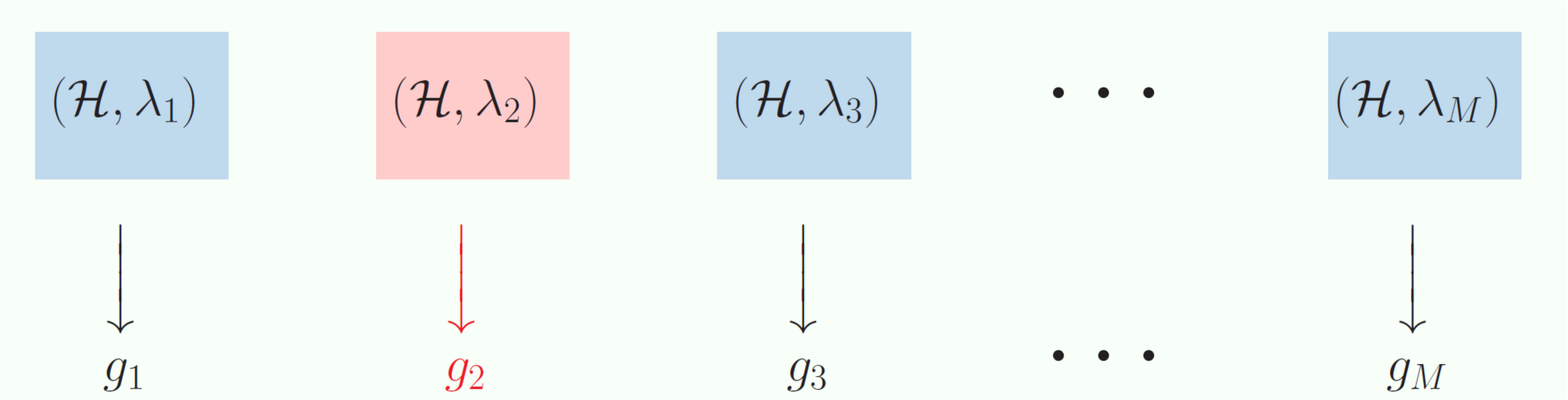
Pick the model with the minimum validation error: $g^{-}_{m^*}$



# Let’s choice λ

What regularization parameter to use? : $\lambda_1, \lambda_2, \ldots, \lambda_M$

This is an special case of *model selection* over M models


$(\mathcal{H}, \lambda_1)$
$(\mathcal{H}, \lambda_2)$
$(\mathcal{H}, \lambda_3)$
$\cdots$
$(\mathcal{H}, \lambda_M)$
$g_1$
$g_2$
$g_3$
$\cdots$
$g_M$

Picking a model amount to choosing the optimal λ

# Error bound for the best

Now $E_{val}(g_{m^*}^-)$ is not an unbiased estimator for $E_{out}(g_{m^*}^-)$

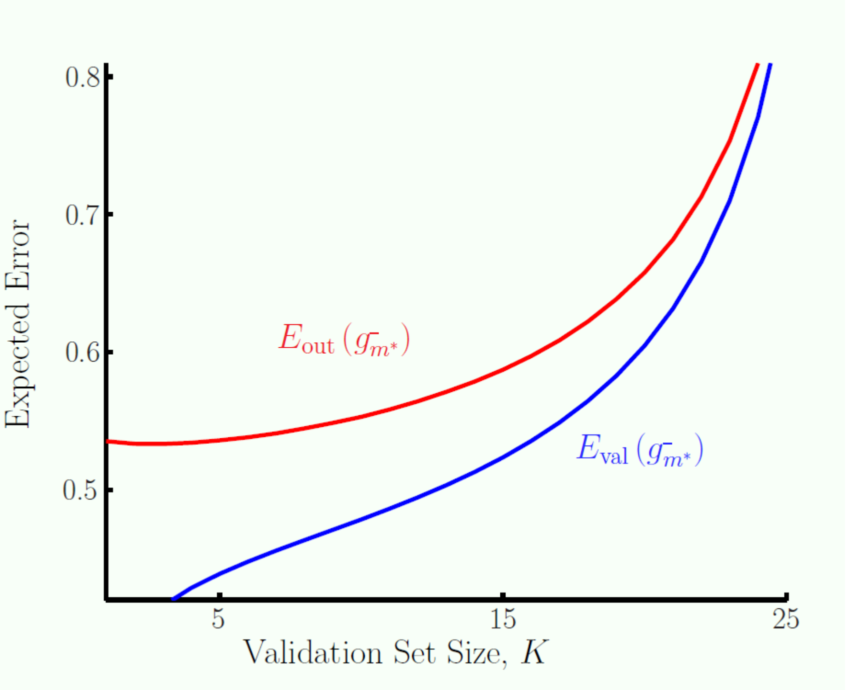
$E_{val}(g_{m^*}^-)$ will be an optimistic estimator

. . . . . becouse we choose one of the M finalist with the minimum error.

How good is the generalization error for this entire process of model selection using validation ?

According to the Hoeffding's inequality for finite hypothesis

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

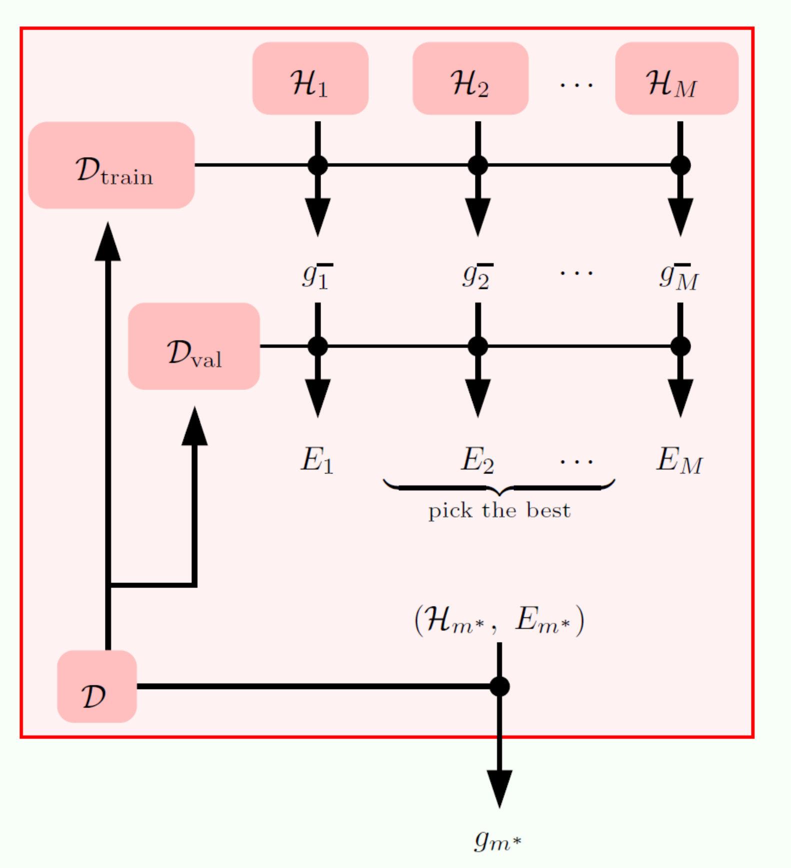


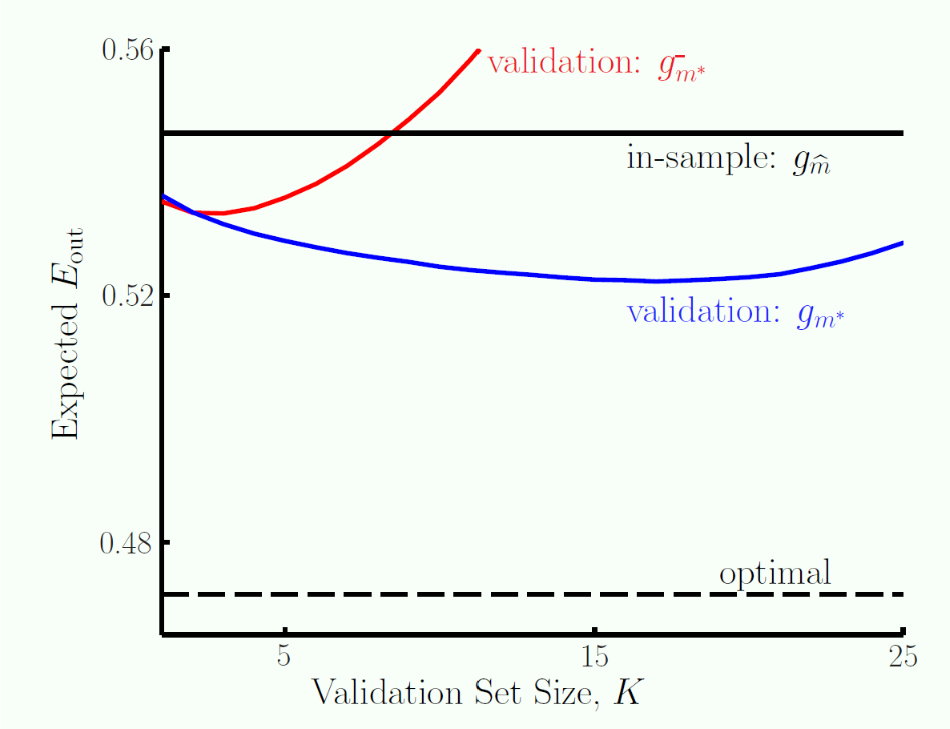
Simulated example of optimistic bias of the validation error when using a validation set for the model selected. ( see the book for the details)

Learning curves

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

# Comparing $E_{in}$ and $E_{val}$ for Model Selection





Simulated example: Curves for model selection from two hypothesis $(\mathcal{H}_2, \mathcal{H}_5)$ using a range of K-values

It can be appreciated how the use of validation for model selection provides in general the hypothesis with the minimum expected out-of-sample error

For k small the validation error is better than in-sample selection

# The Dilema when choosing K

Validation relies on the following chain of reasoning:

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

Small K Large K

What to do ?

Taking K small we get a very good agreement between the two estimates of the out-of-sample error BUT in this case the validation error though still unbiased will have a very large variance.

Solution: Let's fix K=1 and use CROSS-VALIDATION

CROSS-VALIDATION estimates the validation error as an average of validation errors.

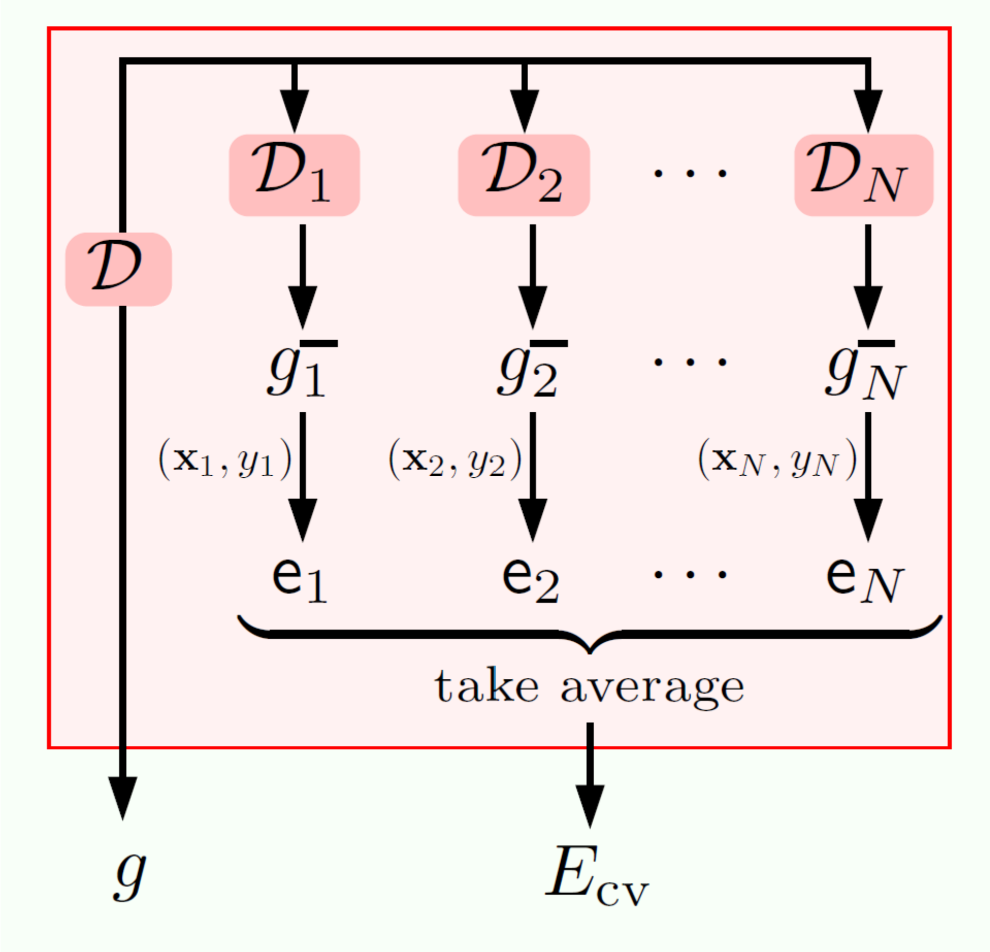
# Leave-One-Out

Let fix K=1 and compute N different models using a validation set with a single element

$$E_{\mathrm{cv}} = \frac{1}{N}\sum_{i=1}^{N} E_{\mathrm{val}}(g_i^-)$$

Result: $E_{\mathrm{cv}}$ is an unbiased estimate of

$$\bar{E}_{out}(N-1) = E_{\mathcal{D}_n}[E_{out}(g_n^-)].$$



According to the learning curves $E_{\mathrm{out}}(g) \le E_{\mathrm{cv}}(g)$

BUT how stable is the estimation using K=1?: GOOD ENOUGH !! (experimental result) )

# Cross-Validation is computationally intensive

### N **epochs of learning each of size** N-1

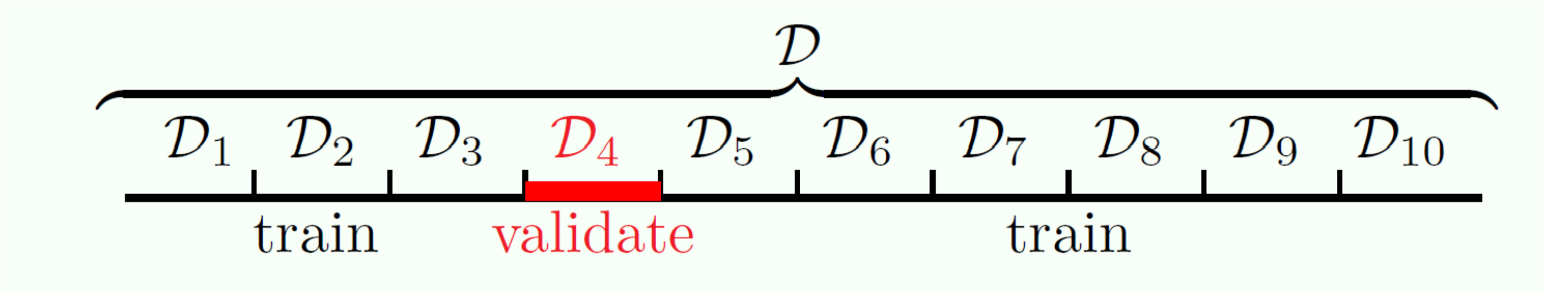
- For linear regression is possible to compute analitically this error

$$\mathbf{w}_{\text{reg}} = (Z^T Z + \lambda \mathrm{I})^{-1} Z^T \boldsymbol{y}$$

$$E_{\text{cv}} = \frac{1}{N} \sum_{n=1}^{N} \left( \frac{y_n - \hat{y}_n}{1 - H_{nn}(\lambda)} \right)^2$$

$$H(\lambda) = Z(Z^T Z + \lambda \mathrm{I})^{-1} Z^T$$

- **A general alternative is** V**-fold Cross-Validation** (V ∈ [5 , 10] )



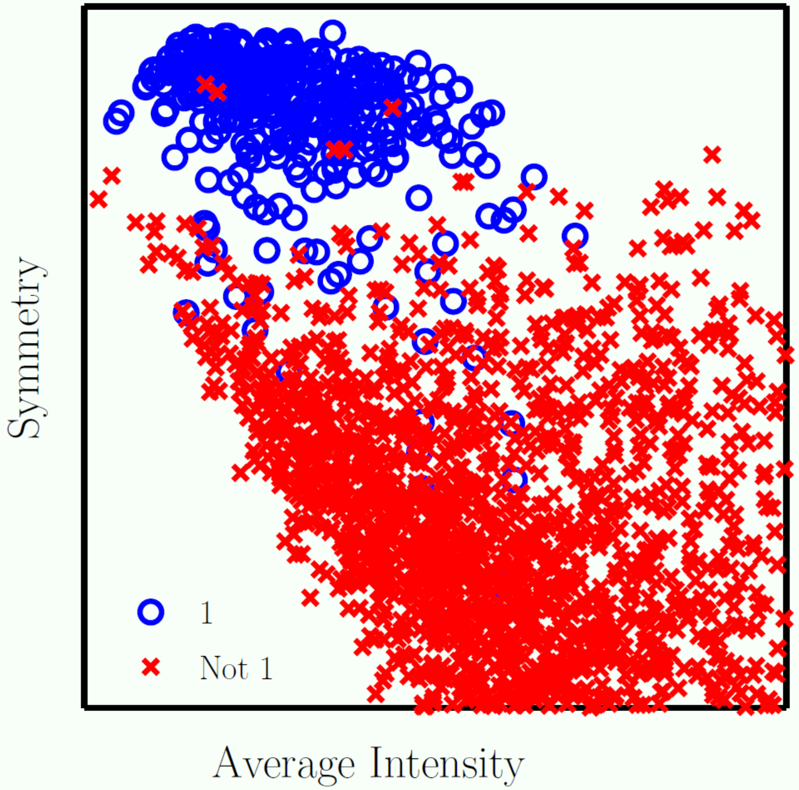
# Summary of Results

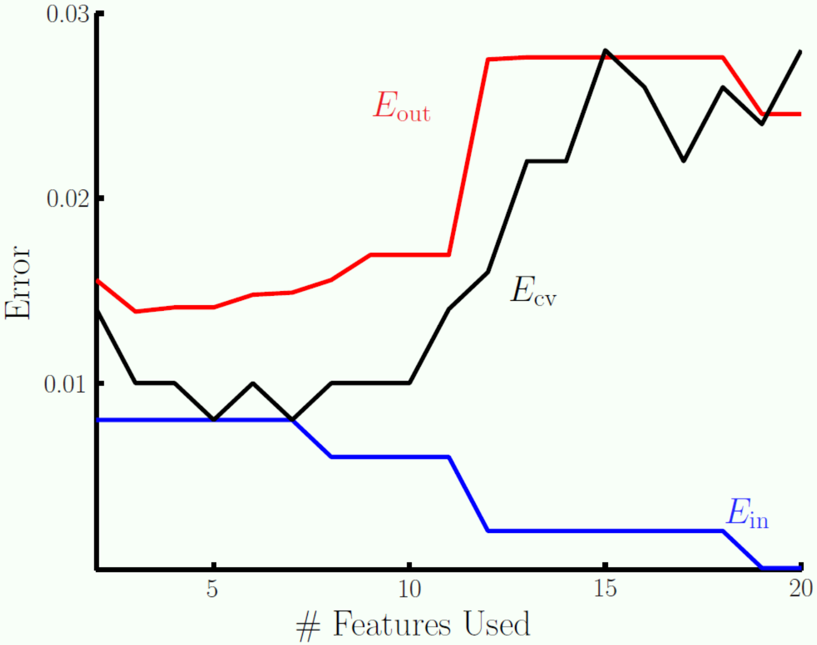
- **Noise** (stochastic o deterministic) affects learning adversely, leading to overfitting

- **Regularization** helps to prevent overfitting by constraining the model, reducing the impact of the noise, while still giving us flexibility to fit the data.

- **Validation** and **Cross-Validation** are useful techniques for **model selection** and **$E_{out}$ estimation.**

# Example: digit classification

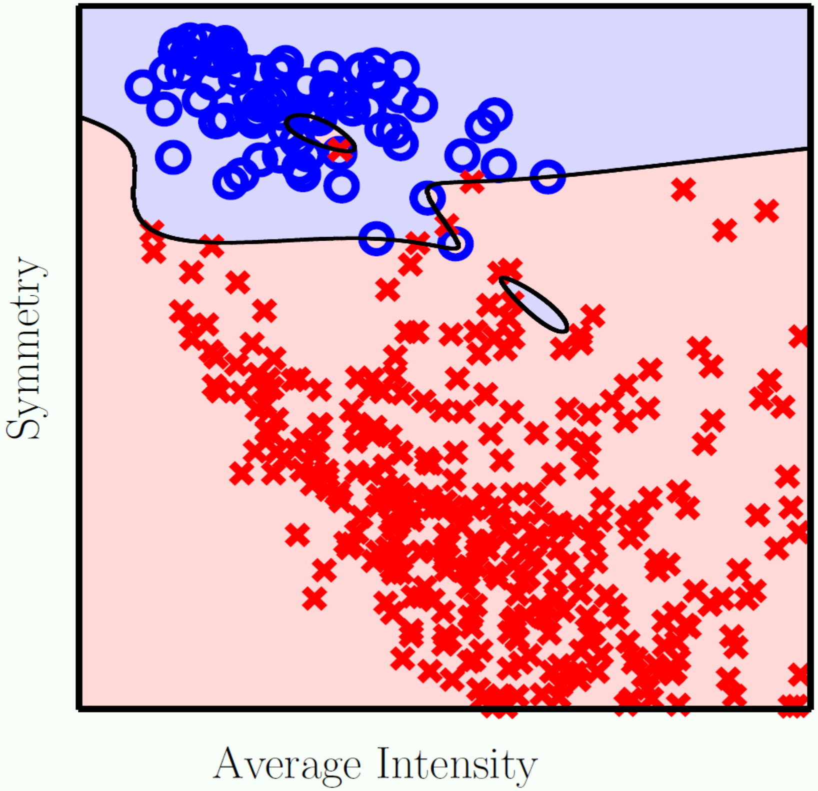




$$\mathbf{x} = (1, x_1, x_2)$$

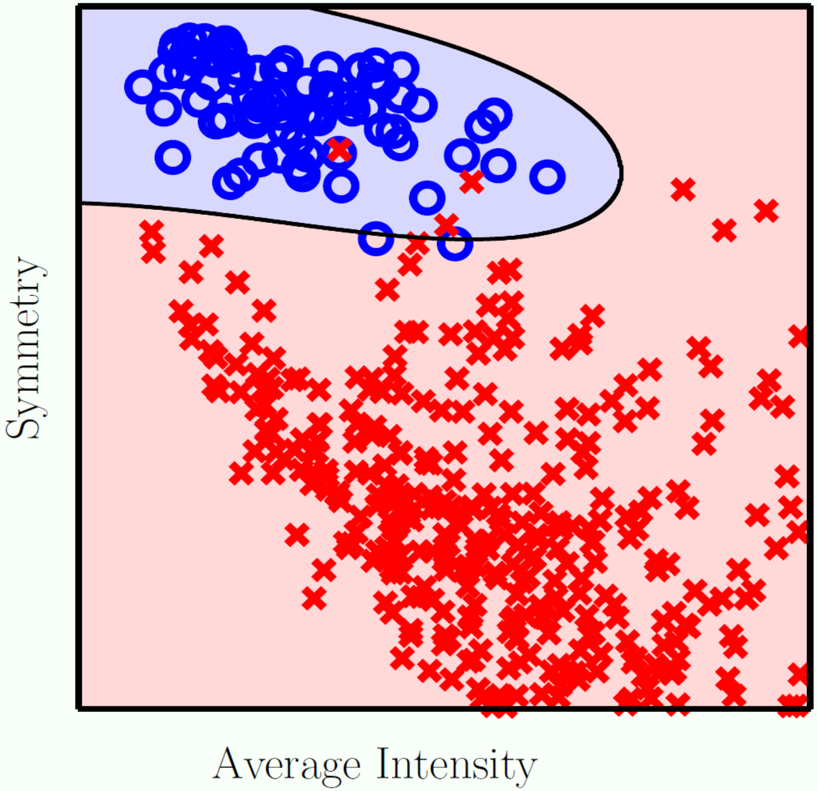
$$\mathbf{z} = \underbrace{(1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3, \ldots, x_1^5, x_1^4 x_2, x_1^3 x_2^2, x_1^2 x_2^3, x_1 x_2^4, x_2^5)}_{\text{5th order polynomial transform} \longrightarrow \text{20 dimensional non linear feature space}}$$

# Example: digit classification



no validation (20 features)

$E_{\text{in}} = 0\%$
$E_{\text{out}} = 2.5\%$



cross validation (6 features)

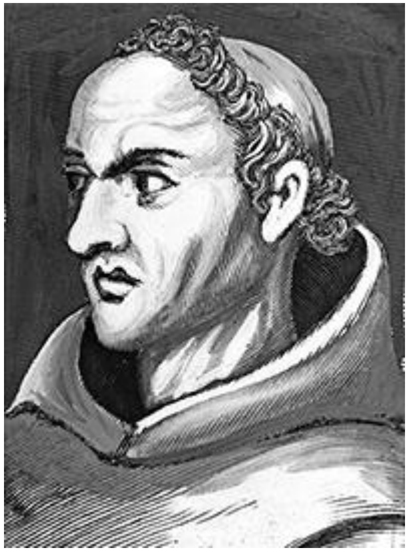
$E_{\text{in}} = 0.8\%$
$E_{\text{out}} = \mathbf{1.5\%}$

# Three Learning Principles

# We will discuss…..

- Occam’s razor: pick a model carefully
- Sample Bias: generate data carefully
- Data Snooping: handle the data caerfully

# Occam's razor



Entities should not be multiplied beyond necessity ''Occam's razor'' principle attributed to William of Occam c. 1280–1349

We should seek simpler models over complex ones and optimize the tradeoff between model complexity and the accuracy of model's description of the training data

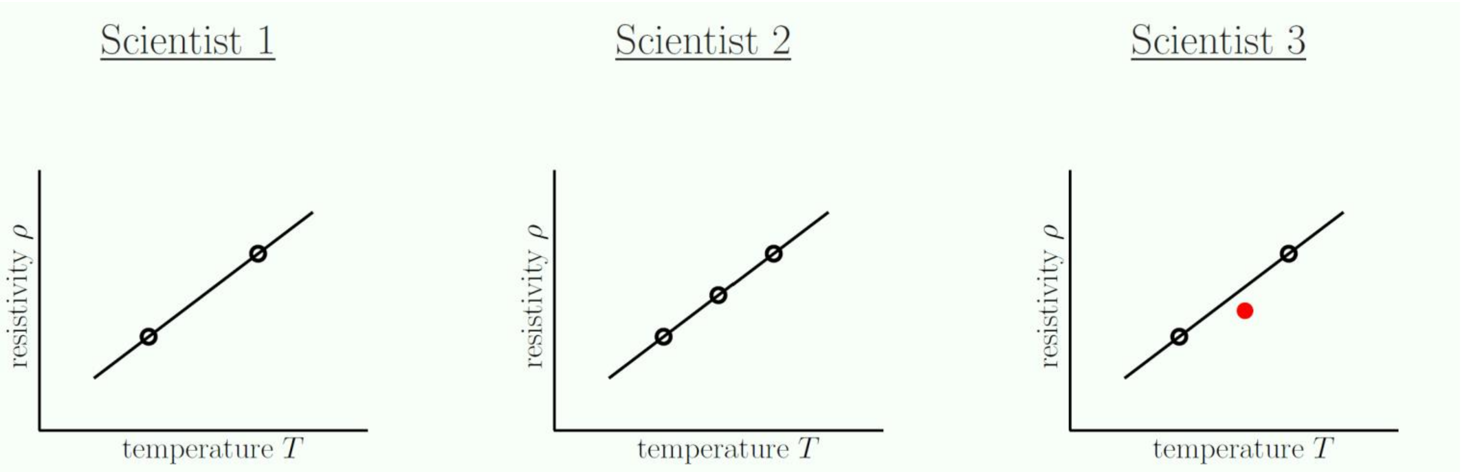
# Why Simpler is Better… ?

- Mathematically : many arguments ( lower VC dimension, less capability to fit noise, etc,etc)

Simple is better because you will be more surprised when you fit the data

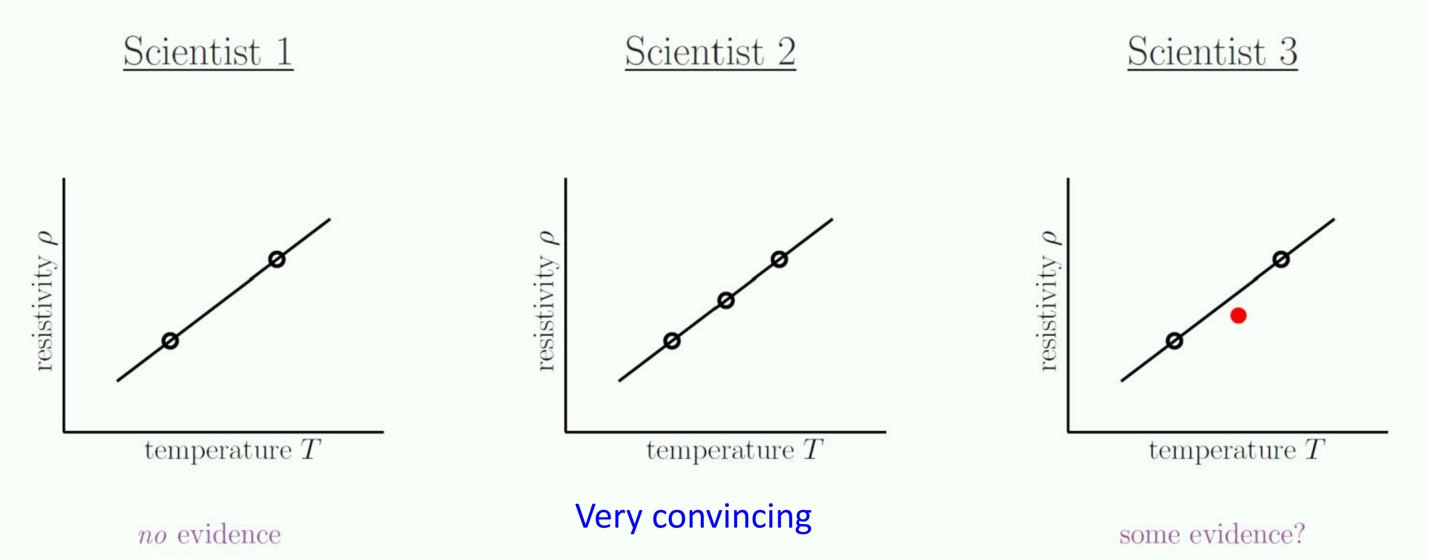
If something unlikely happens, it is very significant when it happens.

A scientific experiment

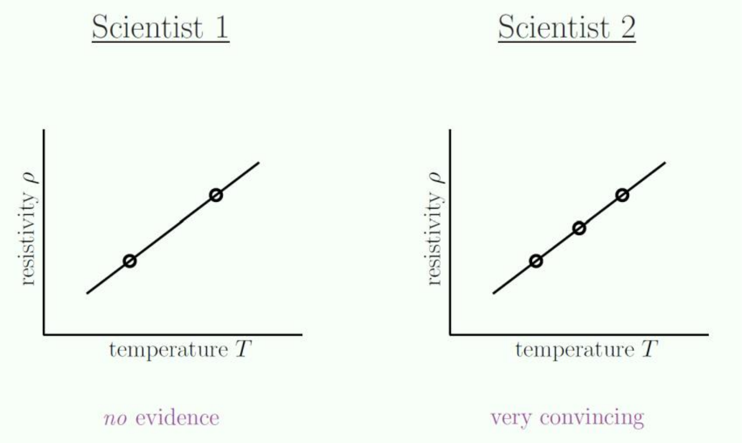


Who provides most evidence to the hypothesis "$\rho$ is linear in $T$" ?

# Scientific Experiment



Very convincing



**Axiom of Non-Falsifiability.**
*If an experiment has no chance of falsifying a hypothesis, then the result of that experiment provides no evidence one way or the other for the hypothesis.*

# Falsification and $m_{\mathcal{H}}(N)$

- If $\mathcal{H}$ shatter $\mathbf{x}_1$, $\mathbf{x}_2$,…, $\mathbf{x}_N$

  … don't be surprised if you fit the data

- If $\mathcal{H}$ doesn't shatter $\mathbf{x}_1$, $\mathbf{x}_2$,…, $\mathbf{x}_N$, and the target values are uniformly distributed,

$$\mathbb{P}[\text{falsification}] \geq 1 - \frac{m_{\mathcal{H}}(N)}{2^N}$$

A good fit is surprising with simpler $\mathcal{H}$ (small $m_{\mathcal{H}}(N)$) and hence more significant.

GOING BEYOND OCCAM'S RAZOR . . .

We may opt for 'a simpler fit than possible', namely an imperfect fit of the data using a simple model over a perfect fit using a more complex one. The reason is that the price we pay for a perfect fit in terms of the penalty for model complexity may be too much in comparison to the benefit of the better fit.

# Sampling Bias in Learning

If the data is sampled in a biased way, learning will produce a similarly biased outcome.

. . . or, make sure the training and test distributions are the same.
**You cannot draw a sample from one distribution and make claims about another distribution**

Example.1: Consider the data set of customer loan from a Bank. ¿where is the bias when used to build a rule for new credit appointment ?

Example.2: Consider the data set of historical values of the current trading companies to select companies in which to invest ¿where is the bias?

# Data Snooping

If a data set has affected any step in the learning process, it cannot be fully trusted in assessing the outcome.

. . . or, estimate performance with a completely uncontaminated test set
. . . and, **choose** $\mathcal{H}$ **before** looking at the data

**Example** : Consider a 8 years data set of the daily changes USD/EURO. We want predict UP/DOWN.

Before fitting a model:
1.- We normalize the fully data set.
2.- We separate the last two years from the data set as Test Set.
3.- We fit a model
4.- The fitted model works very well on the data set,
5.- But when used with new data the prediction error was very high,

is there any rational explanation?

# Data Snooping is a <u>Subtle</u> Happy Hell

- The data looks linear, so I will use a linear model, and it worked.
  - If the data were different and didn't look linear, would you do something different?

- Try linear, it fails; try circles it works.
  - If you torture the data enough, it will confess.

- Try linear, it works; so I don't need to try circles.
  - Would you have tried circles if the data were different?

- Read papers, see what others did on the data. Modify and improve on that.
  - If the data were different, would that modify what others did and hence what you did?
  - the data snooping can happen all at once or sequentially by different people

- Input normalization: normalize the data, now set aside the test set.
  - Since the test set was involved in the normalization, wouldn't your *g* change if the test set changed?