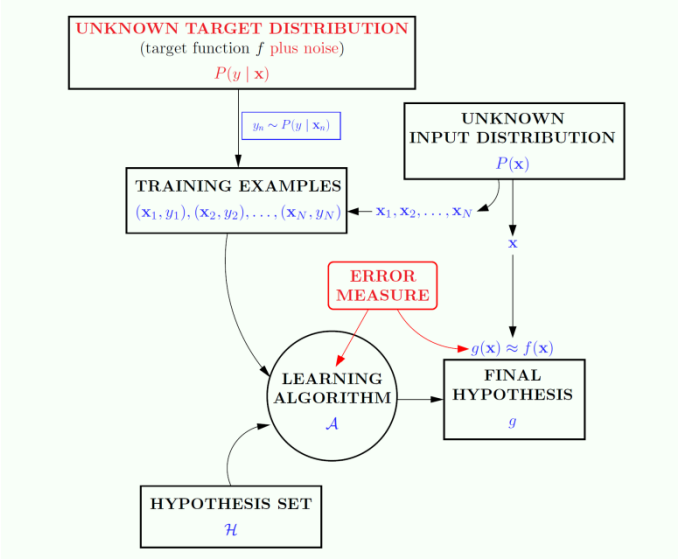
Full Summary

Theory



$$\mathbb{P}(\mathcal{D}\text{: } |E_{\text{out}}(h) - E_{\text{in}}(h)| > \epsilon) \leq 2e^{-2\epsilon^2 N}$$

Uniform Convergence

for any $\epsilon > 0$ *and* $\forall g \in \mathcal{H}$

$$\mathbb{P}(\mathcal{D}\text{: } |\mathrm{E}_{in}(g) - \mathrm{E}_{out}(g)| > \epsilon) < 2|\mathcal{H}|e^{-2\epsilon^2 N}$$

With probability at least $1 - \delta$,

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{1}{2N}\ln\frac{2|\mathcal{H}|}{\delta}}$$

Models

Classification: Perceptron

$$h(\boldsymbol{x}) = sign(\boldsymbol{w}^T\boldsymbol{x})$$

Error: 0/1

Algorithm: PLA / Pocket

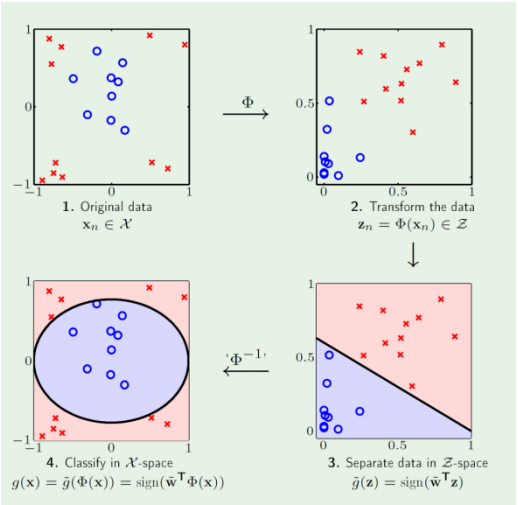
Regression: Lineal

$$h(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$$

Error: quadratic

Algorithm: Linear system using SVD

Adding more features



ERM theory of Generalization:

The Vapnik-Chervonenkis Dimension

$\mathcal{H}$- infinite: the discretization trick

- The discretization trick allows us to have an estimation for the sample complexity inequality on infinite classes

- Example:
  - A modern computer use a 64 bit representation for each scalar.
  - Whether we have to fit functions with only one free parameter, we only have 2^{64} possible values
  - The size of $\mathcal{H}$ now is 2^{64}
  - In the case of d free parameters the size will be 2^{64d}
  - Applying the inequality for finite classes, we obtain a bound the for the sample complexity given by

$$m_{\mathcal{H}}(\varepsilon,\delta) \leq \left\lceil \frac{2}{\epsilon^2} \log \frac{2|\mathcal{H}|}{\delta} \right\rceil = \frac{128d + 2\log(2/\delta)}{\epsilon^2}$$

  - This bound allow us to get a very rough estimate of the required sample complexity in practical situations

- Is there anything better…?

What is the uniform inequality pitfall ?

- Let's remember the simple bound we use:

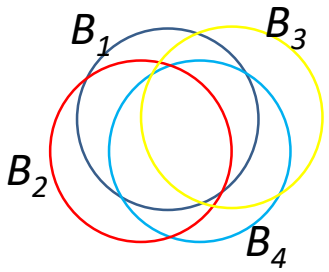
$$P\left(\bigcup_{i=1:|\mathcal{H}|} B_i\right) \leq \sum_{i=1}^{|\mathcal{H}|} P(B_i)$$

- and its consequence

$$P(D: |\mathrm{E}_{in}(g) - \mathrm{E}_{out}(g)| > \epsilon) < 2|\mathcal{H}|e^{-2\epsilon^2 N} \text{ for any } \epsilon > 0$$

- But in most of the cases, $B_i \cap B_j \neq \emptyset$ for almost all (i,j), hence

$$\bigcup_{i=1:|\mathcal{H}|} B_i = \bigcup_{j=1:|\mathcal{V}|} B_j \quad |\mathcal{V}| \leq |\mathcal{H}|$$



- This means, that counting only a few hypothesis could be sufficient!
- A better bound for the efective number of hypothesis in $\mathcal{H}$ is needed
- The Vapnik-Chervonenkis dimension is the answer !!

VC Generalization Bound

- The VC generalization bound is

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{N}\log\frac{4((2N)^{d_{VC}}+1)}{\delta}}$$

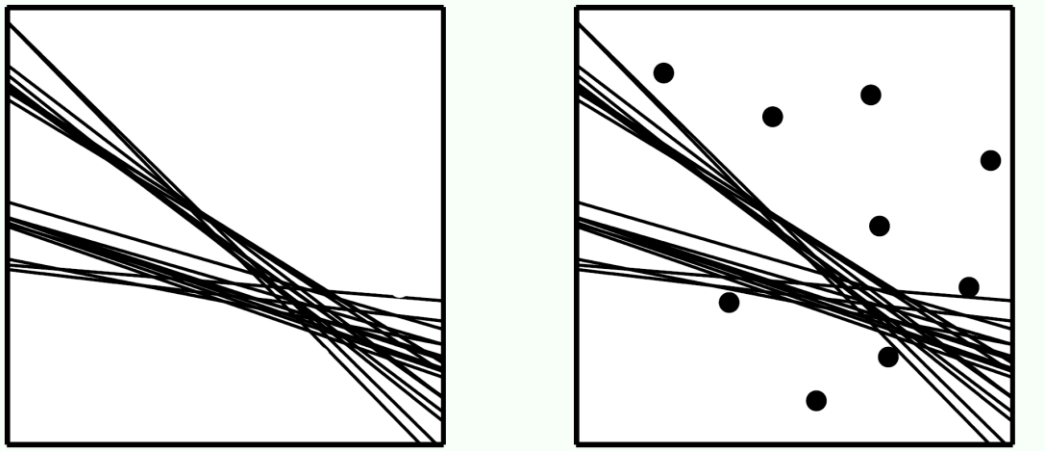
or equivalently $$E_{out}(h) \leq E_{in}(h) + \mathcal{O}\left(\sqrt{d_{VC}\frac{\log N}{N}}\right)$$

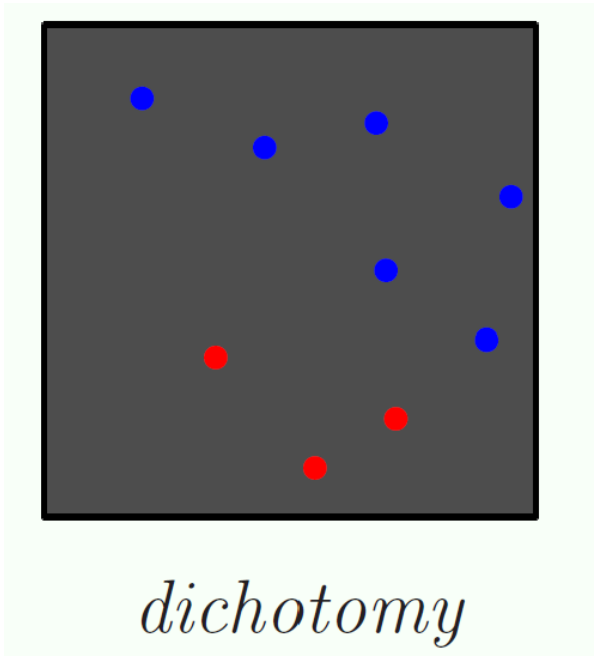
- This shows that for d_{VC} finite and N >> 0, generalization is guaranteed
- As conclusion any model can be considered either Good model or Unknown model
  - Good models: $we\ can\ obtain\ a\ good\ generalization$
  - Unknown models: d_{VC} is infinite (no answer in VC theory)

Measuring the diversity of $\mathcal{H}$

- We need a way to measure the diversity of $\mathcal{H}$
- Here we focus on binary {-1,+1} target functions and finite sample of points

- The approach is combinatorial :
  - Consider a sample of fixed size N
  - Explore if $\mathcal{H}$ can implement ALL possible functions (labeling) on THESE N points
  - Evaluate for all N values





The Growth Function

That is the effective number of function in the class

- The Growth Function $m_{\mathcal{H}}$: Given a sample size N and a class $\mathcal{H}$, $m_{\mathcal{H}}$ return the **maximum number of binary patterns** generated by $\mathcal{H}$ on N points.

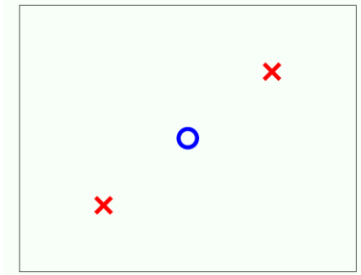
$$m_{\mathcal{H}}(N) = \max_{x_1,\cdots,x_N} |\mathcal{H}(x_1, x_2, \cdots, x_N)|$$

where | | represents number of elements in the set

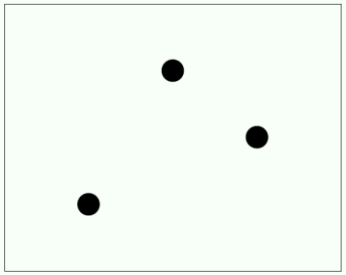
- The **maximum** is computed on all possible samples of size N
- In general $m_{\mathcal{H}}(N) \leq 2^N$
- When $m_{\mathcal{H}}(N) = 2^N$ we say that $\mathcal{H}$ shatter the set $\{x_1, x_2, \cdots, x_N\}$
- It is independent of $\mathcal{P}$, and therefore a worst-case analysis

Growth function: example-1

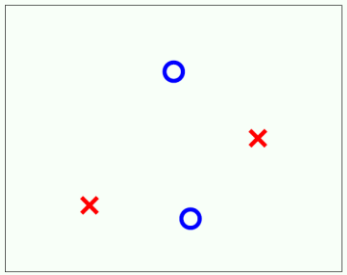
- Let be $\mathcal{H}$ the class of 2D-perceptron



Cannot implement



Can implement all 8



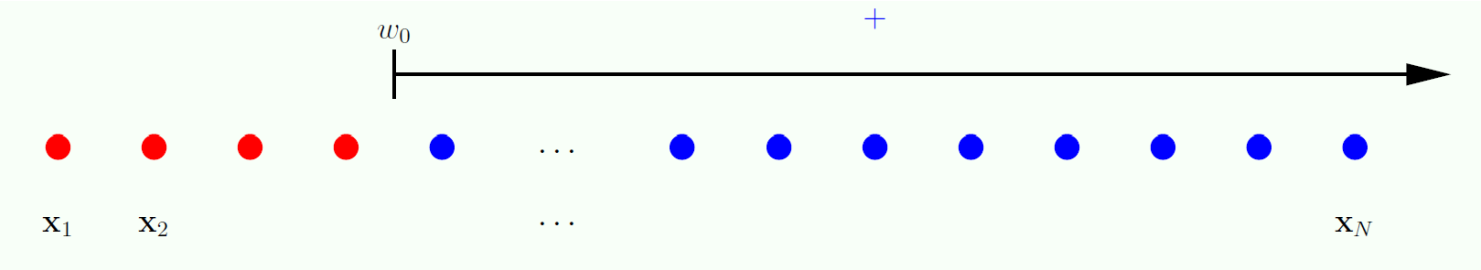
Can implement at most 14

- What is the value of $m_{\mathcal{H}}(N)$?

$$m_{\mathcal{H}}(2) = 4 = 2^2$$
$$m_{\mathcal{H}}(3) = 8 = 2^3$$
$$m_{\mathcal{H}}(4) = 14 < 2^4$$

- Let be $\mathcal{H}$ the perceptron class (binary linear predictors) and $\mathcal{X} = \mathbb{R}^3$
  - Can be shattered a sample of 2 points? , 3 points ?, 4 points ?, etc
- Can you guess any rule for points in $\mathbb{R}^k$?

Growth function: example-2

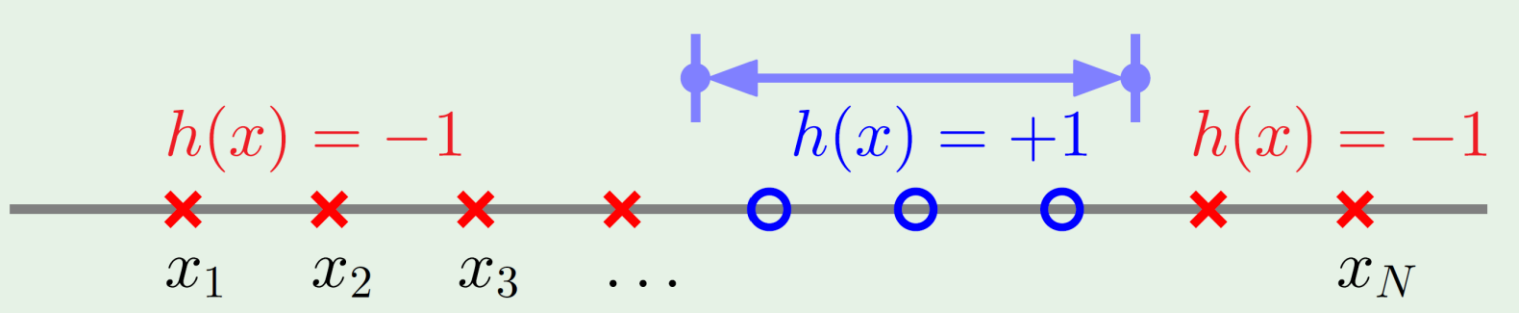


- Let be $\mathcal{H}$ the class of h: $\mathbb{R} \rightarrow \{-1,+1\}$ (Positives Rays)
  $h(x) = \text{sign}(x - w_0)$

- Let consider a sample of N points from $\mathbb{R}$.
  - Question: What is the value of $m_{\mathcal{H}}(N)$?
  - Answer: {N+1,N,N-1}, which of them ?

- How many points can be shattered ?

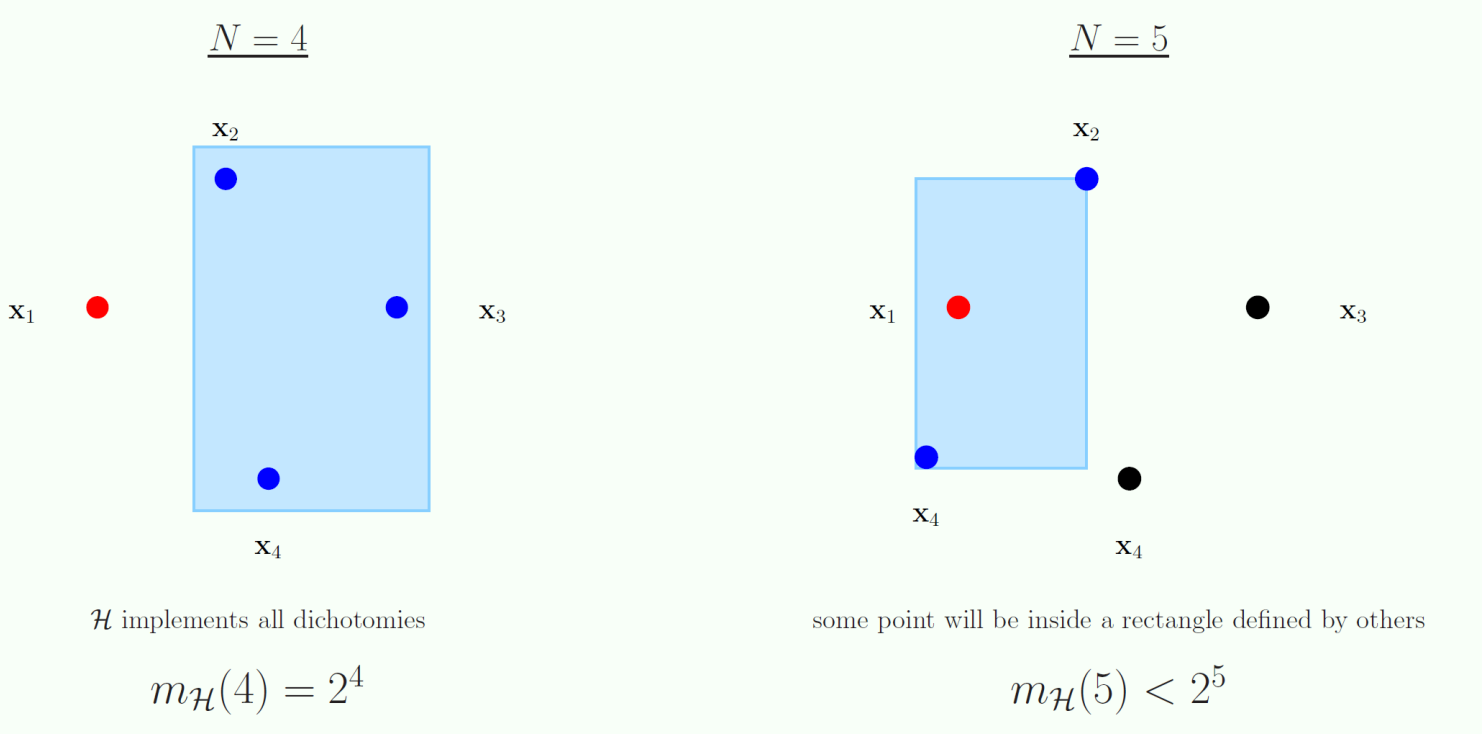
Growth function: example-3



- Let be $\mathcal{H}$ the class of functions h: $\mathbb{R} \rightarrow$ {-1,+1} (Intervals)
- $h_{a,b}(\mathrm{x}) = \begin{cases} +1 & \text{if } \mathrm{x} \in [a,b] \\ -1 & \text{if } \mathrm{x} \notin [a,b] \end{cases}$
- Let consider a sample of N points from $\mathbb{R}$.
- Now: $m_{\mathcal{H}}(N) = \binom{N+1}{2} = \frac{1}{2}N^2 + \frac{1}{2}N + 1$ Why?
- How many points can be shattered ?

Growth function: example-4

- Let $\mathcal{H}$ be the class of positive rectangles



To compute $m_{\mathcal{H}}(5)$ is NOT easy!!

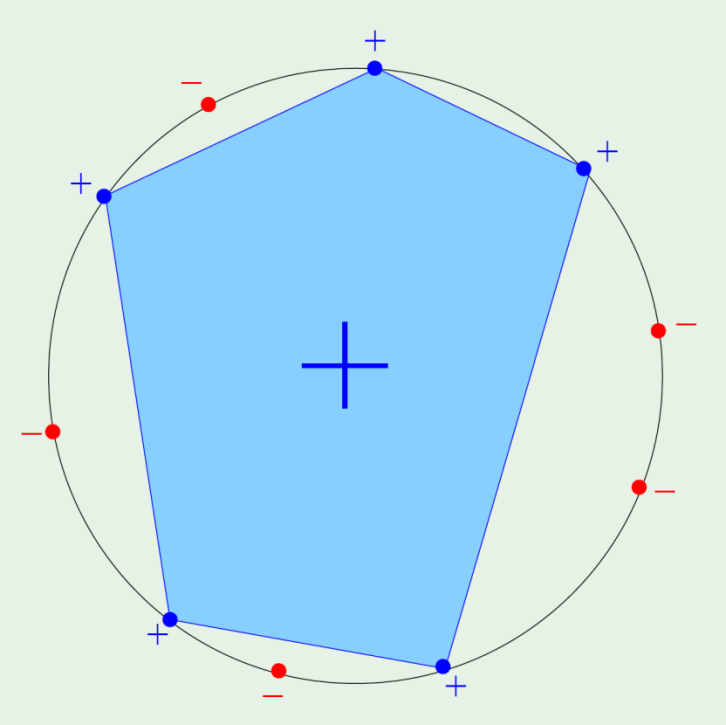
Growth function: example-5

- Let $\mathcal{H}$ be the class of convex set
- Consider the case where all point lies on a circle

$\mathcal{H}$ is set of $h: \mathbb{R}^2 \rightarrow \{-1, +1\}$

$h(\mathbf{x}) = +1$ is convex

$m_{\mathcal{H}}(N) = 2^N$



Growth Function and Generalization

- Let's have a look to the new bound we get:

$$P(\mathcal{D}: |E_{in}(h) - E_{out}(h)| > \epsilon) \leq 4m_{\mathcal{H}}(2N)e^{-\frac{N\epsilon^2}{8}}$$

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{N}\log\frac{4m_{\mathcal{H}}(2N)}{\delta}}$$

- This expressión provides a better bound but required to compute the growth function.

- A constant upper bound on $m_{\mathcal{H}}(N)$ will also solve the problem
- The new bound **is not** a direct replacement of $|\mathcal{H}|$ by $m_{\mathcal{H}}(N)$!!

Break Point

- It is not practical to try to compute $m_{\mathcal{H}}(N)$ for every hypothesis set we use, an upper bound will be sufficient.

- **Break Point Concept:**

  if for some value k, $m_{\mathcal{H}}(k) < 2^k$, then k is a **break point** for $\mathcal{H}$

  - **That is, $\mathcal{H}$ CANNOT shatter a sample of size k**

- **Examples:**
  - **Which is the break point for the 2D Perceptron? k=4**
  - **Which is the break point for the Positive Rays? k=2**
  - **Which is the break point for the Interval? k=3**
  - **Which is the break point for the Positive Rectangle? k=5**
  - **Which is the break point for the Convex Set? k=∞ (it doesn`t exist)**

Bounding the Growth Function

- **Main Result (Vapnik&Chervonenkis, 1971)**: if $k\ is\ a\ break\ point\ for\ \mathcal{H}$, then for all N

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

The RHS **is polynomial in N** of degree k-1: $\mathcal{O}(N^{k-1})$

This result says: if $\mathcal{H}$ has a break point : $m_{\mathcal{H}}(N)$ is **polinomia**l in N

if $\mathcal{H}$ has NOT break point : $m_{\mathcal{H}}(N) = 2^N$

- Regarding the generalization bound we replace $\log m_{\mathcal{H}}(N)$ by $\mathcal{O}(k \log N)$.
  - For N >> 0 we can guarantee a good generalization since log(N)/N -> 0

- What happens at the generalization bound when $\mathcal{H}$ has NOT a breakpoint?

Vapnik&Chervonenkis: VC-dimension:

- **Definition:** *The VC dimension of a hypothesis set* $\mathcal{H}$*, denote by* $d_{VC}(\mathcal{H})$ *or simply* d_{VC}*, is the largest value of N for which* $m_{\mathcal{H}}(N) = 2^N$*. If* $m_{\mathcal{H}}(N) = 2^N$ *for all N, then* $d_{VC} = \infty$.

- It can be proved that the main bound can be written as

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{d_{VC}} \binom{N}{i} \leq \begin{cases} N^{d_{VC}} + 1 \\ \left(\frac{eN}{d_{VC}}\right)^{d_{VC}} \end{cases}$$

Two bounds for the growth function

- The d_{VC} value measure the "**effective** " number of parameters associated with $h \in \mathcal{H}$ (Perceptron 2D, d_{VC}=3; in **linear models d_{VC}=d+1**)

VC Generalization Bound

- Combining bounds

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{N}\log\frac{4((2N)^{d_{VC}}+1)}{\delta}}$$

or equivalently $$E_{out}(h) \leq E_{in}(h) + \mathcal{O}\left(\sqrt{d_{VC}\frac{\log N}{N}}\right)$$

- This shows that for d_{VC} finite and N >> 0, generalization is guaranteed
- A conclusion is that there are a division of models in two classes: "Good" models and "Useless" models (in terms of ERM learning)
  - "Good" models: d_{VC} is finite $we\ can\ obtain\ a\ good\ generalization$
  - "Useless" models: d_{VC} is infinite (we can not learn using the ERM rule!!)

Sample Complexity

- Remember: The sample complexity is the minimum number of training examples (N) needed to achieve a certain generalization performance
  - ε, δ *have to be fixed*
  - How fast grows N(ε, δ) indicates how much data is needed to get good generalization.

- Fix $\delta > 0$ and suppose the generalization error to be at most ε

$$\sqrt{\frac{8}{N}\ln\frac{4m_{\mathcal{H}}(2N)}{\delta}} \leq \varepsilon \Rightarrow N \geq \frac{8}{\varepsilon^2}\ln\left(\frac{4m_{\mathcal{H}}(2N)}{\delta}\right) \Rightarrow N \geq \frac{8}{\varepsilon^2}\ln\left(\frac{4\left((2N)^{d_{\mathrm{VC}}}+1\right)}{\delta}\right)$$

  - This is an implicit equation in N, we solve it iteratively

Sample Complexity: An Example

- Example:
  - Suppose d_{VC}=3
  - Assume $\varepsilon = 0.1$, $\delta = 0.1$. How big a data set do we need?

$$N \geq \frac{8}{0.1^2} \ln\left(\frac{4(2N)^3+4}{0.1}\right) \xrightarrow{N=1000} N \geq 21.193 \Rightarrow N \geq 30.000$$

fixed point of the equation

  - For d_{VC}=4, we get $N \geq 40.000$
  - For d_{VC}=5, we get $N \geq 50.000$
  - This suggest the bound should be proportional to d_{vc}

  - A good rule of thumb : $N > 10 \times d_{vc}$

VC means Penalty by Model Complexity

- In most practical situations the sample data set is given, so N is fixed !

- The relevant question now is what performance can we expected given this particular N

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N}\ln\frac{4m_{\mathcal{H}}(2N)}{\delta}} \leq E_{in}(g) + \sqrt{\frac{8}{N}\ln\frac{4\left((2N)^{d_{\mathrm{VC}}}+1\right)}{\delta}}$$

$$\Omega(N,\mathcal{H},\delta) = \sqrt{\frac{8}{N}\ln\frac{4\left((2N)^{d_{\mathrm{VC}}}+1\right)}{\delta}} = \mathcal{O}\left(\sqrt{\frac{d_{VC}\ln N - \ln\delta}{N}}\right)$$

- This term can be seen as a penalty due to the $\mathcal{H}$ complexity.

$$E_{out} \leq E_{in} + \Omega(d_{\mathrm{VC}})$$

VC Bound Quantifies Approximation vs Generalization

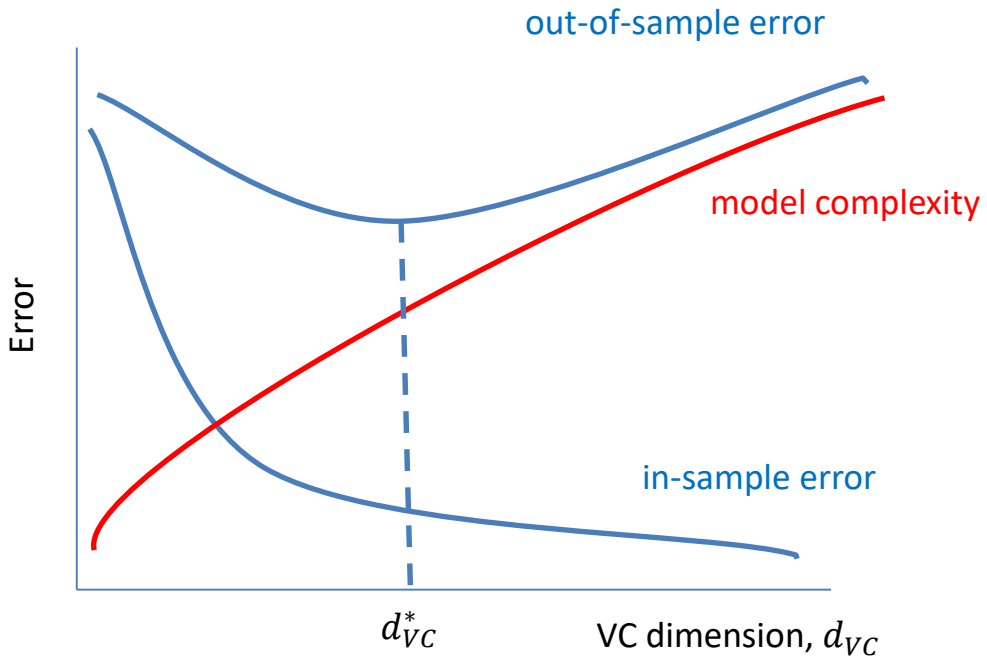
- In fact, we have a tradeoff: More complex models help E_{in} and hurt $\Omega(N, \mathcal{H}, \delta)$

- $d_{\text{vc}} \uparrow \Longrightarrow$ better chance of approximating f ($E_{in} \approx 0$).
- $d_{\text{vc}} \downarrow \Longrightarrow$ better chance of generalizing to out of sample ($E_{in} \approx E_{out}$).

$$E_{out} \leq E_{in} + \Omega(d_{\text{vc}})$$

- **VC analysis only depends on $\mathcal{H}$.**
  - Independent of f, $\mathbb{P}(\mathcal{X})$, $\mathcal{A}$ (learning algorithm)
  - Mainly applicable to classification and regression problems
  - Nevertheless, for square loss a better insight is given by the Bias-Variance tradeoff.
  - Quite loose bound

Penalty by Model Complexity



- The figure shows the fitting error vs the VC dimension
- A tradeoff, using the out-of-sample error, attains a minimum at some intermediate value d_{VC}^*
- This is a generalization to the finite case tradeoff !!

Model Complexity (d_{vc}) ↑ → E_{in} ↓ → better chance of approximating f

Model Complexity (d_{vc}) ↓ → $E_{\text{out}} - E_{\text{in}}$ ↓ → better chance of good generalization

Summary of the VC Bound

- If $d_{VC}(\mathcal{H})$ is finite $\Longleftrightarrow$ The class $\mathcal{H}$ is "PAC" learnable

- The VC bound is independent of : f, $\mathbb{P}(\mathcal{X})$, $\mathcal{A}$
  - **(Binary target function, Input distribution, Learning Algorithm)**

- The VC dimension give us a measure of the complexity of the class $\mathcal{H}$
  - The higher the complexity the bigger the training set for a fixed error

- The VC dimension of a class $\mathcal{H}$ is related to the "effective" number of free parameters of its elements.

- The VC analysis was developed for to 0-1 loss function (classification)
  - But it can be extended to real-valued loss functions (regression)

How assess our fitting ?

$$E_{\text{out}}(g) \leq E_{\text{in}}(g) + \Omega(N, \mathcal{H}, \delta)$$

is good to guide the training process BUT is useless if we want to get an accuracy forecast of E_{out}.

- In real problems a precise estimate of E_{out} is what the customer is expecting to have.

- The best formula is to challenge our trained hypothesis with absolutely new examples NEVER SEEN BEFORE. This is called a TEST SET
  - The samples of the test set MUST be i.i.d samples from the same probability distribution used in training
- Let us call the error on the test set E_{test}.
- We use E_{test} as an estimator of E_{out}

Why E_{test} should be a good estimator of E_{out} ?

- The answer is in the simple Hoeffding inequality
  - Now we only have one hypothesis and the Hoeffding inequality is very tighter when N increase

$$P(|E_{test}(g) - E_{out}(g)| > \epsilon) \leq 2e^{-2N\epsilon^2}$$

- Example: for 1000 examples of test, E_{test} will be within $\pm 5\%$ of E_{out} with probability $\geq 98\%$

- In addition, the test set estimation is not biased. This means independent of E_{in}

- But nothing is free, there is a price to pay for using a test set
  - We loss training data → Higher in-sample error

NLT- Discussion

- How does the feature transform affect to the PLA VC-bound?
- If we honestly fix the transfom before seeing the data, then $d_{VC}(\mathcal{H}_\Phi) = d_{VC}(\mathcal{H})$ at least with probability 1-δ
- What if we first try separating with lines , fail, and then use the circles?
  - This is equivalent to use a transformation where the original features are keeped and we add the square of all of them.
  - We have increased the dimension of the feature space !!
- What if we explore the data but we do not try any model?
  - Even worst !! Our mind has explored a huge hypothesis space that we must add to the real transformations dimension.
  - Inadvertently, you have decided your data is the problem and not a sample of it !!
- In classification problems if we insist in getting full separability between classes we can be compelled to use high degree transformations
  - Nevertheless, this increase dramatically the feature space dimension and the VC-dimension
- Let analyze in more detail these implications ……

Computation and Generalization

- Let denote by Φ_Q the *Q-th order polynomial transform*
  - $\Phi_4(\boldsymbol{x}) = (1, x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^3, x_2^3, x_1x_2^2, x_2x_1^2, x_1^4, x_2^4, x_1^2x_2^2, x_2^1x_1^3, x_1^1x_2^3)$

- A larger *Q* provides a larger flexibility in terms of the shape of the decision boundary but there is a price to pay.

  1. Computation is an issue because the feature transform Φ_Q maps x (the initial vector) to $d = \frac{Q(Q+3)}{2}$ dimensions, incrementing memory and computational cost.
  2. The VC-dimension can increases till $\frac{Q(Q+3)}{2} + 1$ and the VC-bound can grow significantly
     - For Q=50 the VC-dim is $\frac{Q(Q+3)}{2}$+1 =1326 instead of 3 (initial)
  3. According to the rule: " .. number of samples needed is proportional to the VC-dim", the higher the Q-value the higher (quadratic order) the number of samples we will need to get the same level of generalization error.

- In general when choosing the appropiate dimension for the feature transform, we must use an approximation-generalization tradeoff:

higher d better chance of being linearly separable $(E_{\text{in}} \downarrow)$ and $E_{out} \uparrow$

lower d possibly non linearly separable(E_{in}) and $\text{E}_{out} \downarrow$

What happens when $d_{VC} = \infty$?

- Uniform Learning: From the VC dimension analysis we know that ERM rule is a general learning rule for finite d_{vc}

NONUNIFORM LEARNING

- Now we consider $\mathcal{H} = \bigcup_n \mathcal{H}_n$, $d_{vc}(\mathcal{H}_n) < \infty, n = 1,2,3, \ldots \ldots$
  - This means a class with an infinite VC dimension but defined as the union of a numerable infinity of classes each with $d_{VC} < \infty$
- Example:
  - Class of all polynomials on R. $\mathcal{H} = \bigcup_n \mathcal{H}_n$ where $\mathcal{H}_n$ represents the class of the polynomials of degree n. It's not difficult to show that VCdim$(\mathcal{H}) = \infty$ and VCdim$(\mathcal{H}_n) = \mathrm{n} + 1$

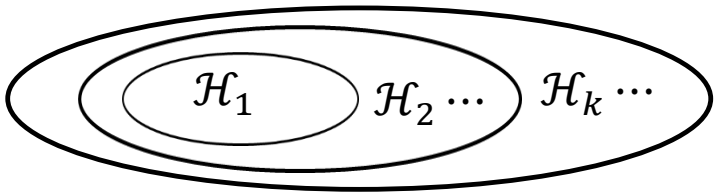
NonUniform learning rule: SRM

$$\Omega(N, \mathcal{H}, \delta) = \mathcal{O}\left(\sqrt{\frac{d_{VC} \ln N - \ln \delta}{N}}\right)$$

- What happens when $\frac{N}{d_{vc}} < 20?$,
  - small number of samples with respect to the number of effective parameters.
- In this case the ERM rule is not a guarantee for learning

A new induction rule is introduced : **Structural RISK Minimization (SRM)**

$$g^* = arg \min_{i=1,2,\ldots} (E_{in}(g_i) + \Omega(\mathcal{H}_i))$$



$$d_{vc}(\mathcal{H}_1) \leq d_{vc}(\mathcal{H}_2) \leq \cdots \leq d_{vc}(\mathcal{H}_k) \leq \cdots$$

SRM

1. Select a nested sequence of hypothesis set
2. Estimate *g* from each set of the sequence

SRM Implementation Criteria

- Keeps the model complexity fixed and minimize empirical error
- Keeps the empirical error constant (small) and minimize VC dimension

Valid for approaches that minimize the true error rather than empirical

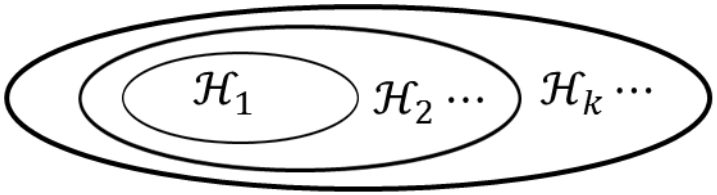
INDUCTION RULES: SUMMARY

Uniform Learning (Minimizing Empirical Error)

- If $d_{VC}(\mathcal{H})$ is finite $\Longleftrightarrow$ $\mathcal{H}$ is agnostic-PAC learnable
- The VC bound is independent of : f, $\mathbb{P}(\mathcal{X})$, $\mathcal{A}$

SRM Learning Criteria (Nonuniform Learning)

$\Omega(N, \mathcal{H}, \delta) = \mathcal{O}\left(\sqrt{\frac{d_{VC} \ln N - \ln \delta}{N}}\right)$ when $\frac{N}{d_{vc}} < 20$ it does not a good guarantee for $E_{out} \approx 0$



$$d_{vc}(\mathcal{H}_1) \leq d_{vc}(\mathcal{H}_2) \leq \cdots \leq d_{vc}(\mathcal{H}_k) \leq \cdots$$

$$g^* = arg \min_{i=1,2,\ldots} (E_{in}(g_i) + \Omega(\mathcal{H}_i))$$

SRM Implementation Criteria

- **Keeps model complexity fixed and minimize empirical error**
- **Keeps empirical error constant (small) y minimize VC dimension**

Sample size depends on the function

Another look at E_{out}

Bias-Variance Tradeoff

- BIAS-VARIANCE decomposition

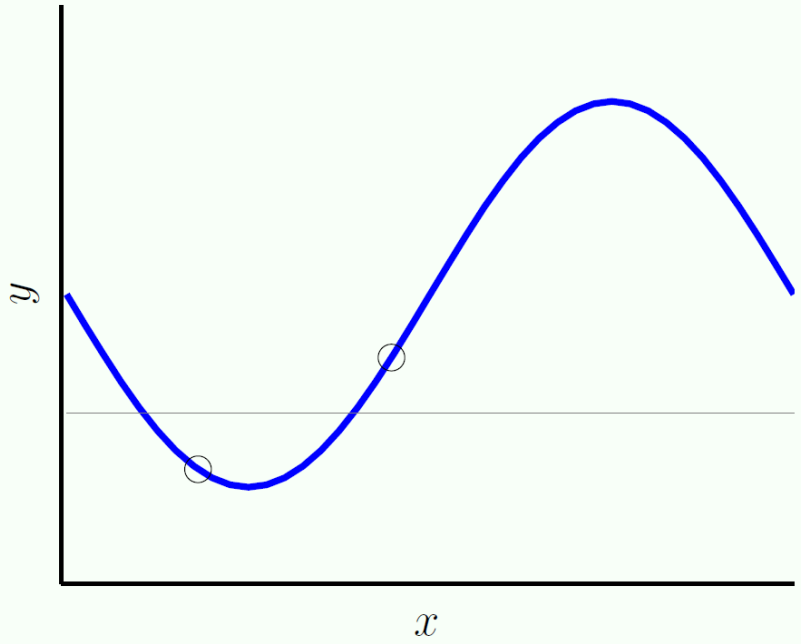
$$E_{\text{out}}\left(g^{(\mathcal{D})}\right) = \mathbb{E}_{\boldsymbol{x}}\left[\left(g^{(\mathcal{D})}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2\right]$$

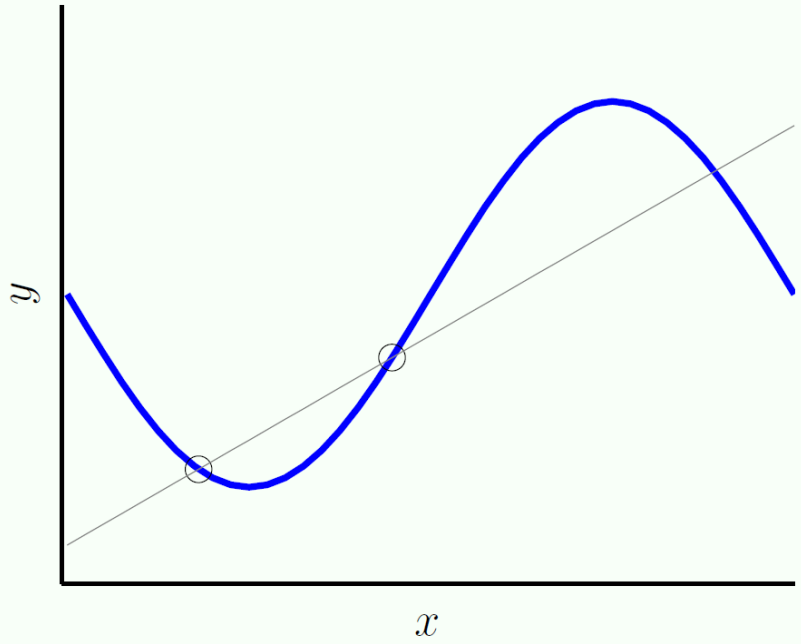
  - $\mathbb{E}_{\boldsymbol{x}}$ denotes the expected value with respect to $\boldsymbol{x}$ (based on $\mathbb{P}(\mathcal{X})$)
  - That is the Mean Squared Error (MSE) of $g^{(\mathcal{D})}$

- Bias-variance analysis is based on squared-errors measure, but applies to classification and regression.
- Bias-variance analysis takes into account $\mathcal{H}$ and $\mathcal{A}$
- Different learning algorithms $\mathcal{A}$ can have different E_{out} when applied to the same $\mathcal{H}$!!

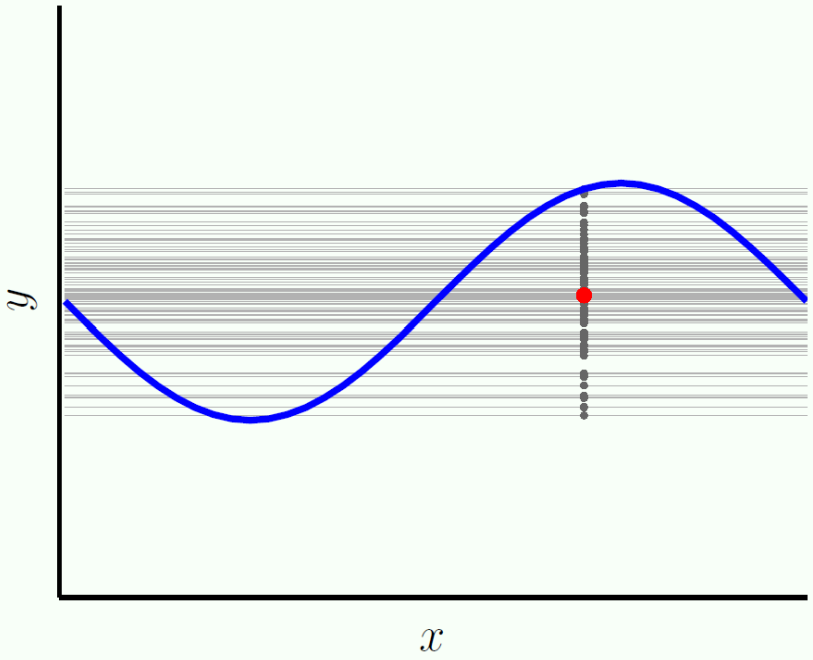
A simple learning problem

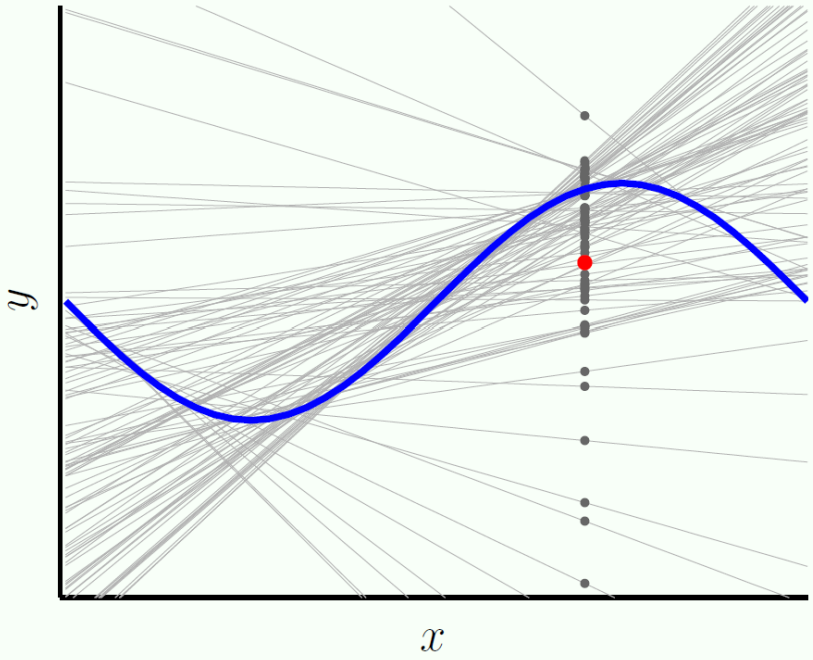
- 2 data points. 2 hypothesis sets
- $\mathcal{H}_0: h(x) = b$
- $\mathcal{H}_1: h(x) = ax + b$





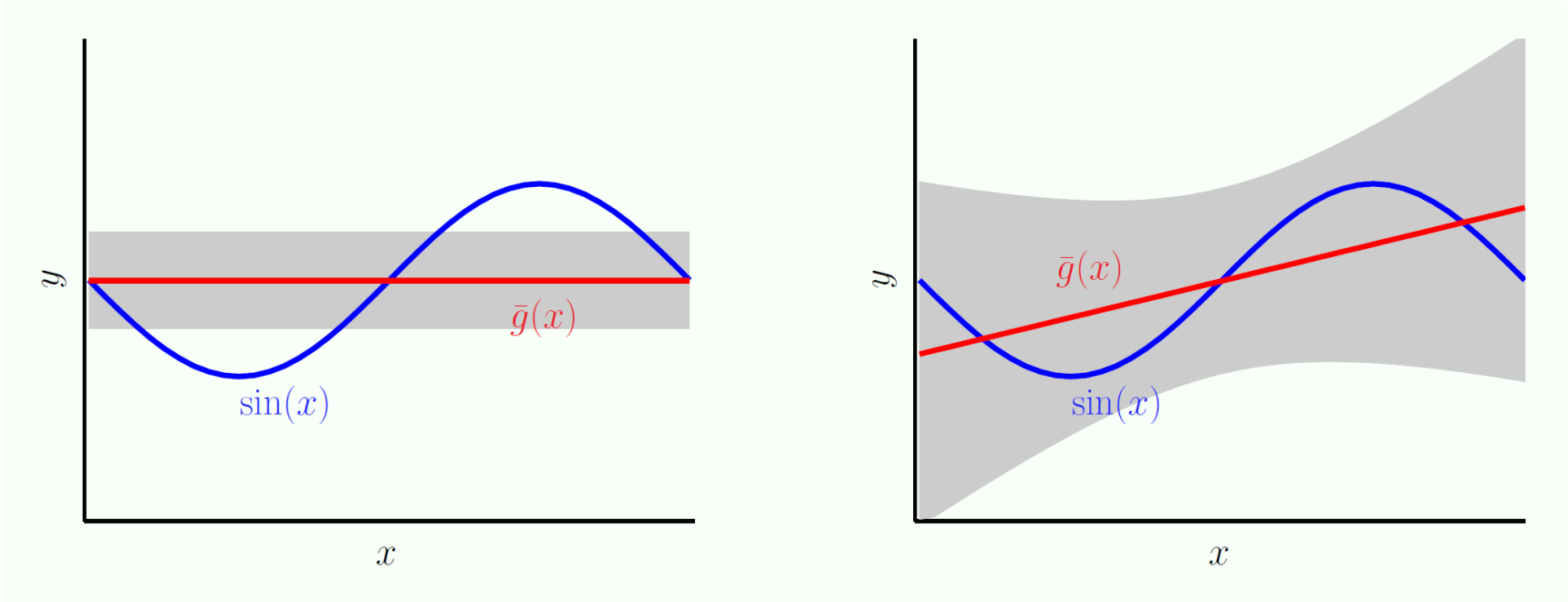
Let repeat the experiment multiples times …





- For each data set $\mathcal{D}$, you get a different $g^{\mathcal{D}}$.
- So, for a fixed $\mathbf{x}$, $g^{\mathcal{D}}(\mathbf{x})$ is random value, depending on $\mathcal{D}$.

What's Happening on Average?



We can define

$g^{\mathcal{D}}(\mathbf{x})$ ← random value, depending on $\mathcal{D}$

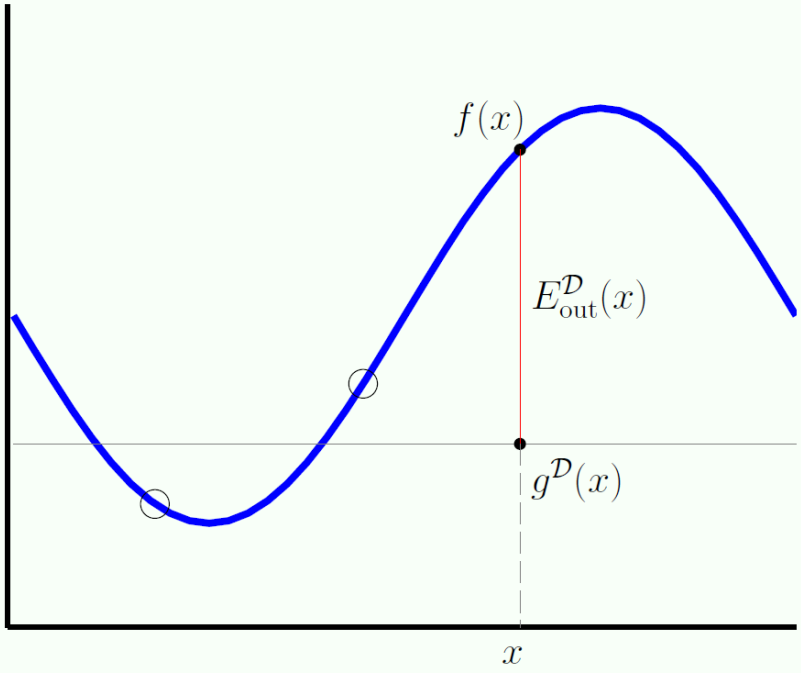
$$\bar{g}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}[g^{\mathcal{D}}(\mathbf{x})]$$

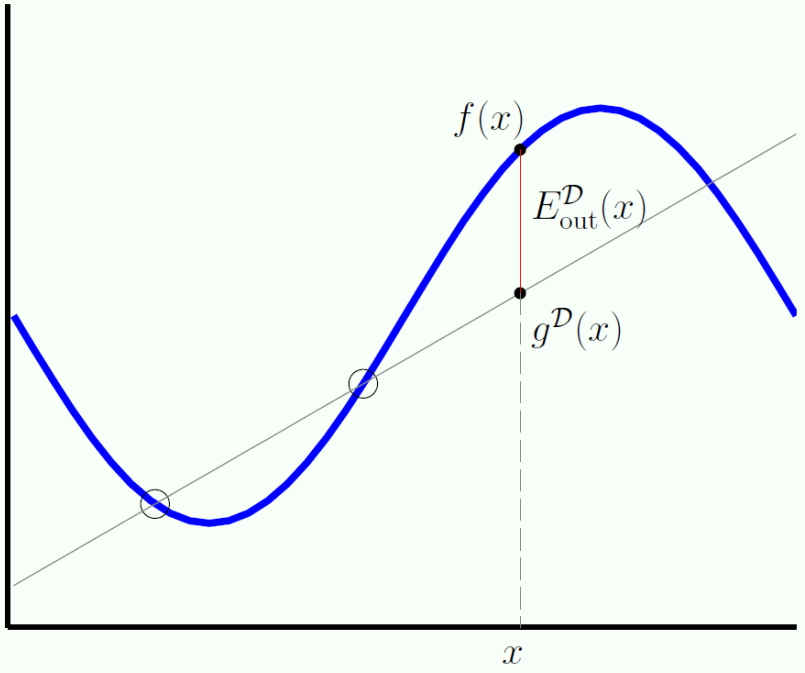
$$\approx \frac{1}{K}\left(g^1(\mathbf{x}) + g^2(\mathbf{x}) + \cdots + g^K(\mathbf{x}) \right)$$ ← the average prediction on **x**

$$\mathrm{var}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}[(g^{\mathcal{D}}(\mathbf{x}) - \bar{g}(\mathbf{x}))^2]$$

$$= \mathbb{E}_{\mathcal{D}}[g^{\mathcal{D}}(\mathbf{x})^2] - \bar{g}(\mathbf{x})^2$$ ← how variable is the prediction?

E_{out} on Test Point **x** for Data $\mathcal{D}$





$E^{\mathcal{D}}_{out}(\mathbf{x}) = (g^{\mathcal{D}}(\mathbf{x}) - f(\mathbf{x}))^2$ ← squared error, a random value depending on $\mathcal{D}$

$E_{out}(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}\left[E^{\mathcal{D}}_{out}(\mathbf{x})\right]$ ← expected $E_{out}(\mathbf{x})$ before seeing $\mathcal{D}$

Bias-Variance Tradeoff

$$\mathbb{E}_{out} = \mathbb{E}_{\mathcal{D}}[E_{out}^{D}] = \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{x}\left[\left(g^{(\mathcal{D})}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2\right]\right] = \mathbb{E}_{x}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2\right]\right]$$

$$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2\right] = \mathbb{E}_{\mathcal{D}}\left(g^{\mathcal{D}}(\boldsymbol{x})^2\right) - 2\,\mathbb{E}_{\mathcal{D}}\left(g^{(\mathcal{D})}(\boldsymbol{x}) f(\boldsymbol{x})\right) + f(\boldsymbol{x})^2$$

- The term $\mathbb{E}_{\mathcal{D}}\left(g^{(\mathcal{D})}(\boldsymbol{x})\right)$ gives an **average function** that we denote by $\tilde{g}(\boldsymbol{x})$

$$\mathbb{E}_{\mathcal{D}}\left[E_{out}\left(g^{(\mathcal{D})}\right)\right] = \mathbb{E}_{x}\left[\mathbb{E}_{\mathcal{D}}\left(g^{\mathcal{D}}(\boldsymbol{x})^2\right) - 2\,\tilde{g}(\boldsymbol{x})\, f(\boldsymbol{x}) + f(\boldsymbol{x})^2\right]$$

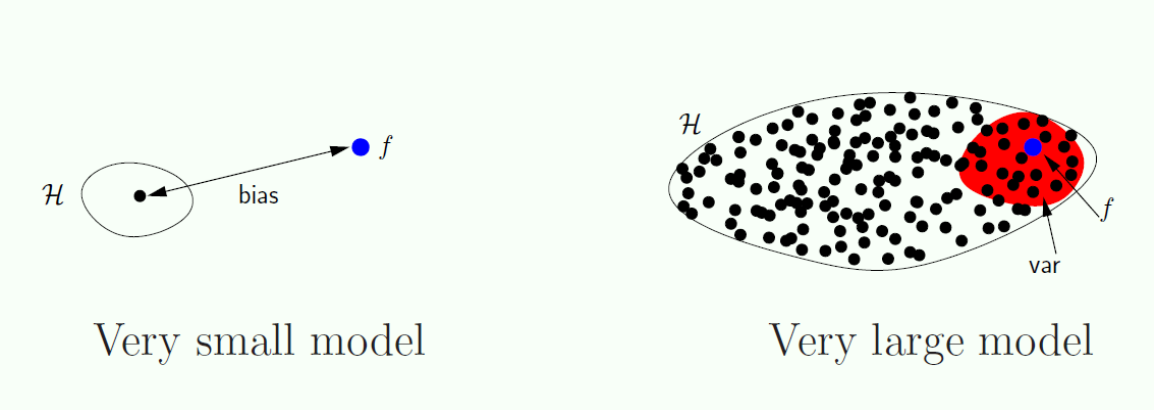
$$= \mathbb{E}_{x}\left[\mathbb{E}_{\mathcal{D}}\underbrace{\left(g^{\mathcal{D}}(\boldsymbol{x})^2\right) - \tilde{g}(\boldsymbol{x})^2} + \underbrace{\tilde{g}(\boldsymbol{x})^2 - 2\,\tilde{g}(\boldsymbol{x})\, f(\boldsymbol{x}) + f(\boldsymbol{x})^2}\right]$$

$\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}(\boldsymbol{x}) - \tilde{g}(\boldsymbol{x})\right)^2\right]$ **variance**$(\boldsymbol{x})$

$(\tilde{g}(\boldsymbol{x}) - f(\boldsymbol{x}))^2$ **bias**$(\boldsymbol{x})$

$$\mathbb{E}_{\mathcal{D}}\left[E_{out}\left(g^{(\mathcal{D})}\right)\right] = \mathbb{E}_{x}[\textbf{bias}(\boldsymbol{x}) + \textbf{variance}(\boldsymbol{x})] = \textbf{bias} + \textbf{variance}$$

Bias-Variance Tradeoff: Comments

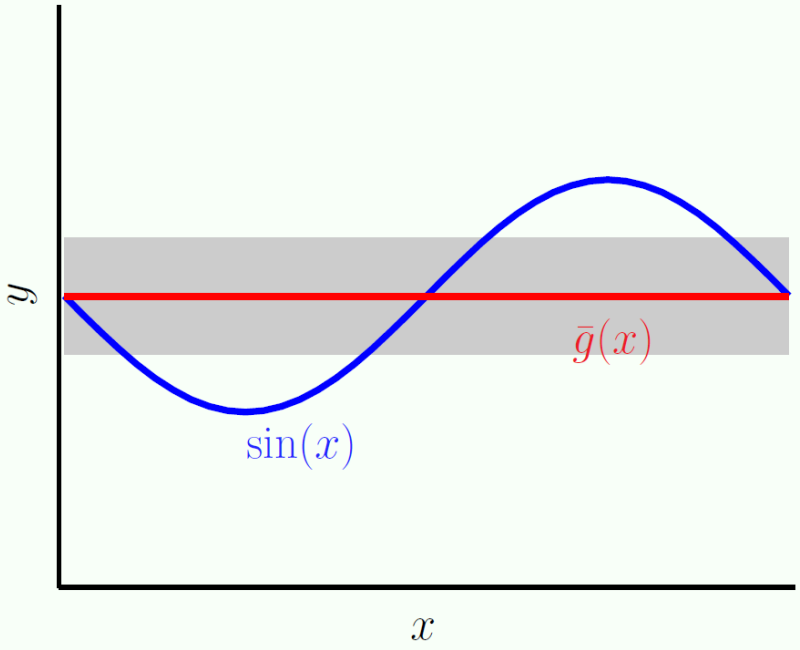


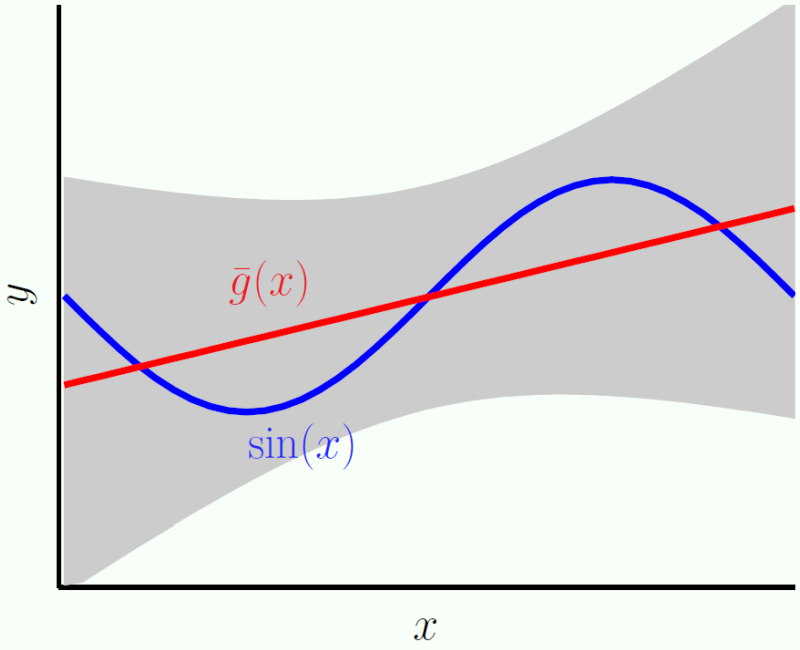
- $\mathbb{E}_{\mathcal{D}}\left[E_{out}\left(g^{(\mathcal{D})}\right)\right] = \sigma^2 + \mathbf{bias} + \mathbf{variance}$ (for noisy signals)
  - σ^2 is the variance of the noise
  - The noise is unavoidable no matter what we do, so our interest remains in bias and variance
  - Unfortunately it is impossible to compute bias and variance. Thus, the bias-variance decomposition is a conceptual tool which is heplful when it comes to developing a model.

- There are two tipical goals when we consider bias and variance:
  - To lower the variance without significatively increase the bias (1)
  - To lower the bias without significatively increase the variance (2)

- **These goals are achieved by different techniques: Regularization(1) , prior knowledge (2)**

Back to $\mathcal{H}_0$ and $\mathcal{H}_1$; and, our winner is . . .





$\mathcal{H}_0$

$\text{bias} = 0.50$

$\text{var} = 0.25$

$E_{\text{out}} = 0.75$ ✓

$\mathcal{H}_1$

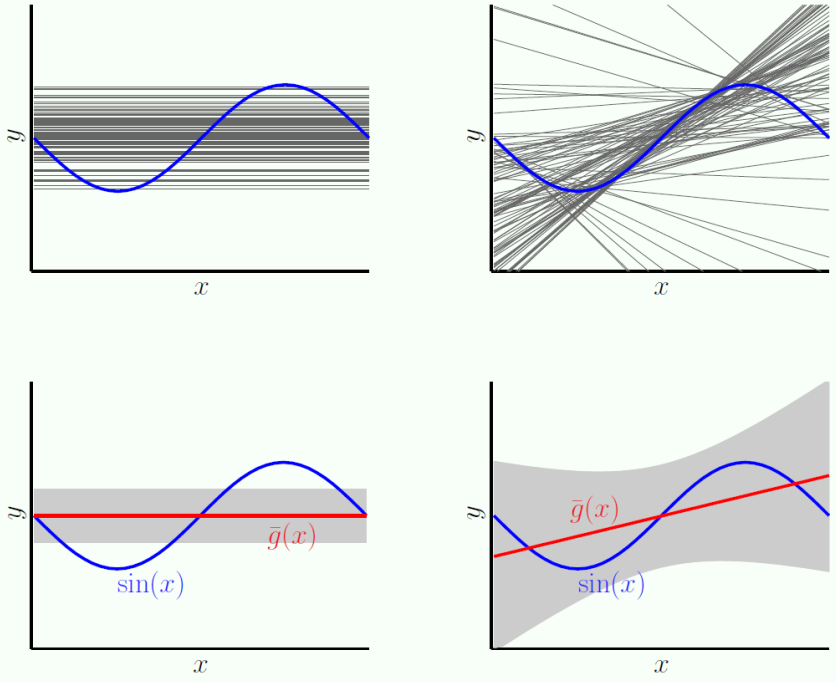
$\text{bias} = 0.21$

$\text{var} = 1.69$

$E_{\text{out}} = 1.90$

Match Learning Power to Data, . . .Not to f

2 Data Points



$\mathcal{H}_0$

bias $= 0.50$;

var $= 0.25$.

$E_{\text{out}} = 0.75$ ✓

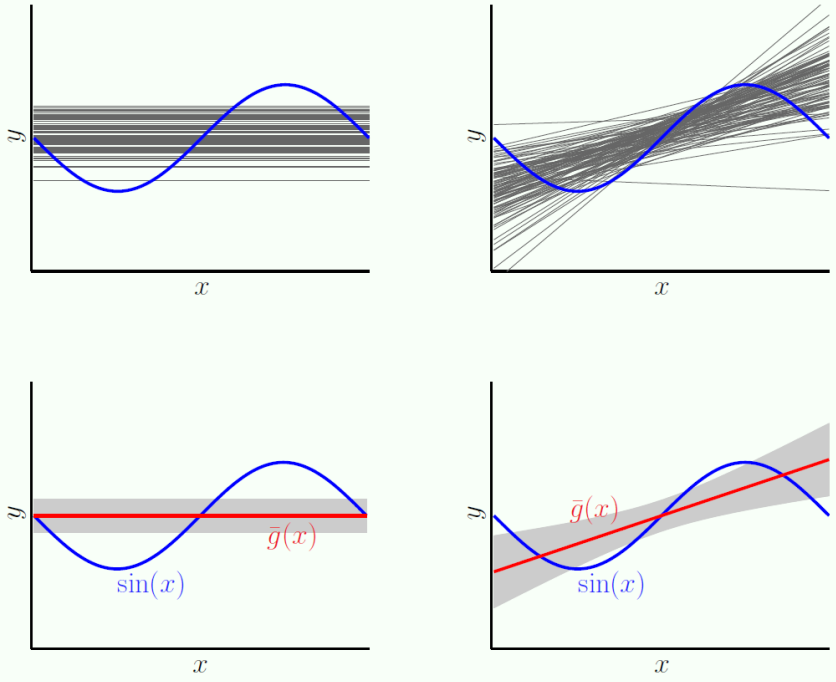
$\mathcal{H}_1$

bias $= 0.21$;

var $= 1.69$.

$E_{\text{out}} = 1.90$

5 Data Points



$\mathcal{H}_0$

bias $= 0.50$;

var $= 0.1$.

$E_{\text{out}} = 0.6$

$\mathcal{H}_1$

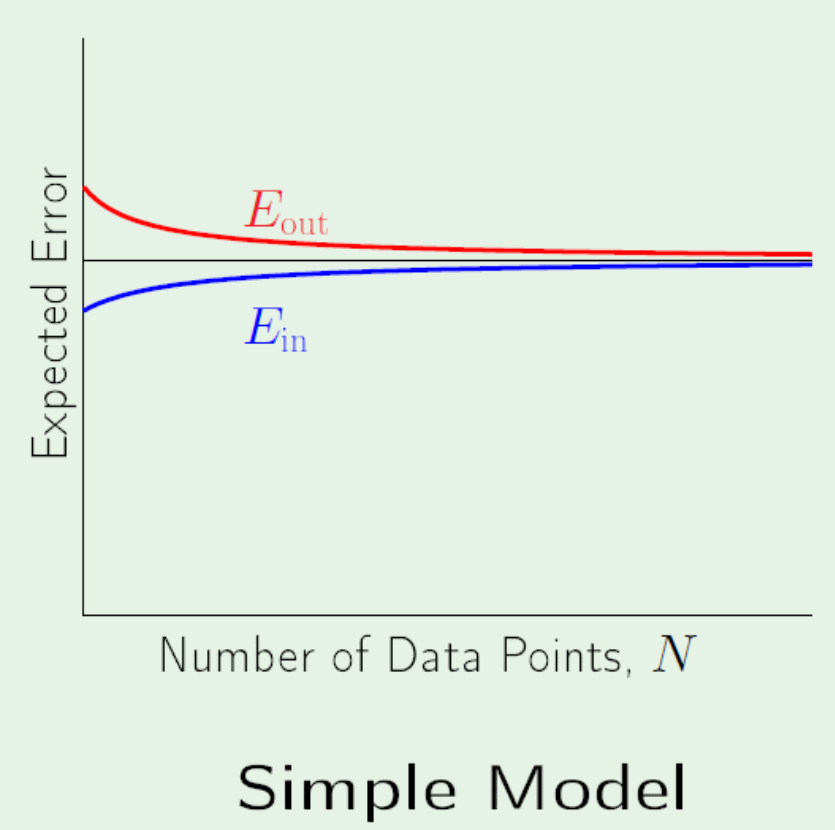
bias $= 0.21$;

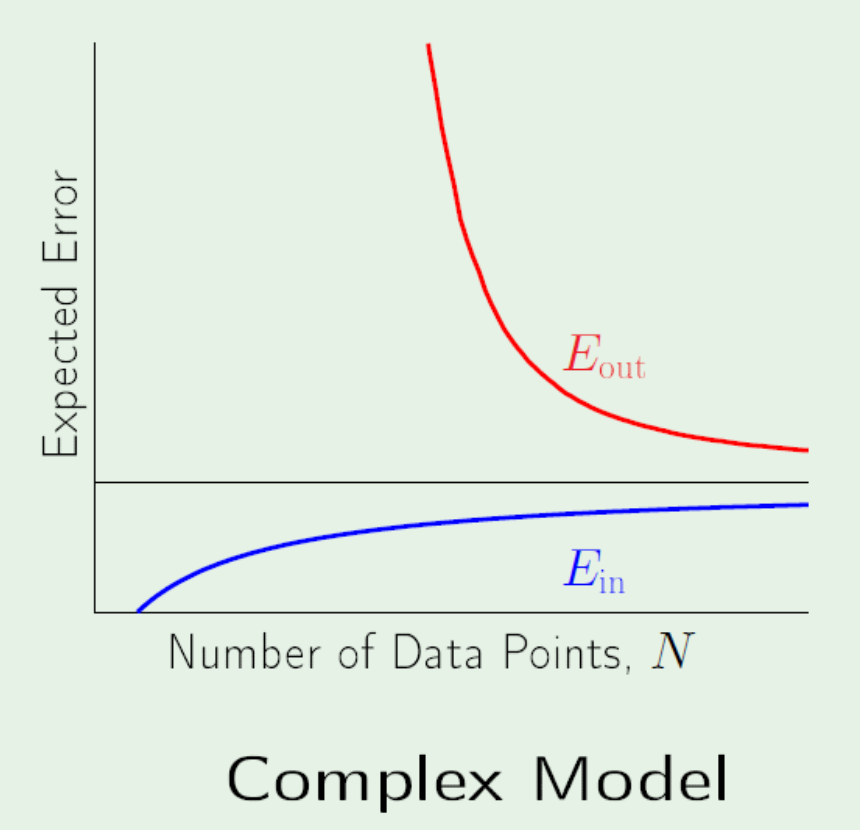
var $= 0.21$.

$E_{\text{out}} = 0.42$ ✓

Learning Curve

- The learning curves summarize the behaviour of the errors $\mathbb{E}_{\mathcal{D}}\left[E_{in}\left(g^{(\mathcal{D})}\right)\right]$ and $\mathbb{E}_{\mathcal{D}}\left[E_{out}\left(g^{(\mathcal{D})}\right)\right]$ when **we vary the size N of the training set** .





The model complexity influence the Expected Error and the speed of convergence

Left: 2nd order polynomial

Right. 10th order polynomial

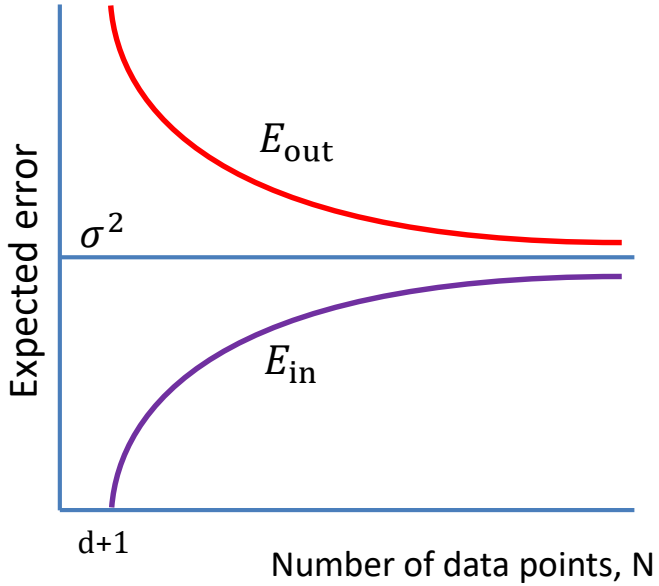
Learning Curve for Linear Regression

- Let now consider the expression for the expected values of $E_{\text{in}}(\mathbf{w}_{\text{lin}})$ and $E_{\text{out}}(\mathbf{w}_{\text{lin}})$

$$\mathbb{E}_{\mathcal{D}}[E_{\text{in}}(\mathbf{w}_{\text{lin}})] = \sigma^2\left(1 - \frac{d+1}{N}\right), \text{ for N} \geq \text{d+1}$$

$$\mathbb{E}_{\mathcal{D}}[E_{\text{test}}(\mathbf{w}_{\text{lin}})] = \sigma^2\left(1 + \frac{d+1}{N}\right) \quad \text{(approx. to } E_{\text{out}}\text{)}$$

The figure shows the linear regression learning curve under the OLS assumptions.



- E_{in} : When N increase the model absorbs as much information as possible with d+1 parameters
- E_{out} : When N increase the out of sample error of the model decreases to the residual noise.
- This behaviour of the learning curve is the expected when the right complexity model has been choosen