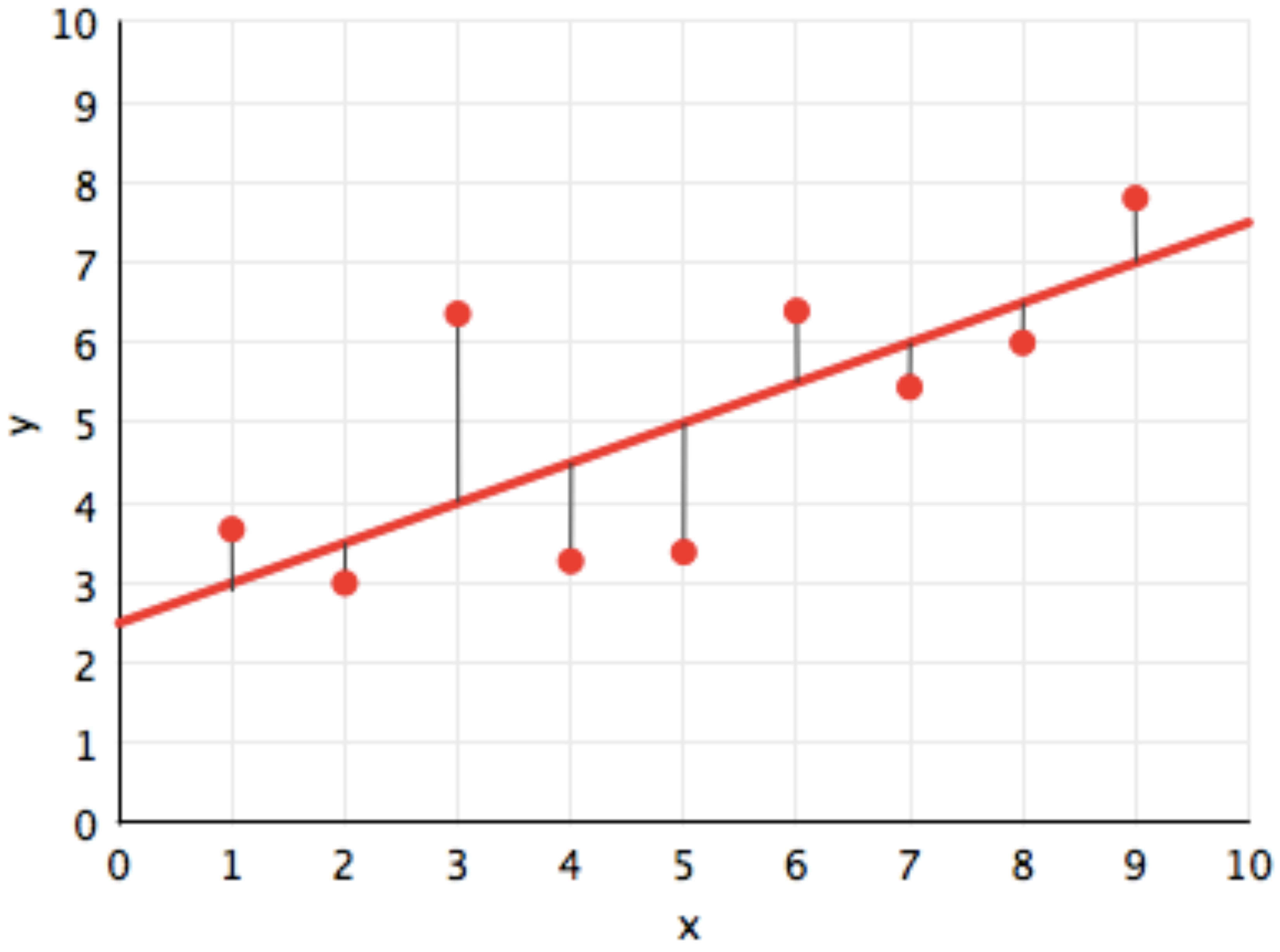
Day 1 Session 2

Learning a Model

Complexity, Validation, and Regularization



RISK: What does it mean to FIT?

Minimize distance from the line?

$$R_{\mathcal{D}}(h_1(x)) = \frac{1}{N} \sum_{y_i \in \mathcal{D}} (y_i - h_1(x_i))^2$$

Minimize squared distance from the line.
Empirical Risk Minimization.

$$g_1(x) = \arg \min_{h_1(x) \in \mathcal{H}} R_{\mathcal{D}}(h_1(x)).$$

Get intercept w_0 and slope w_1.

HYPOTHESIS SPACES

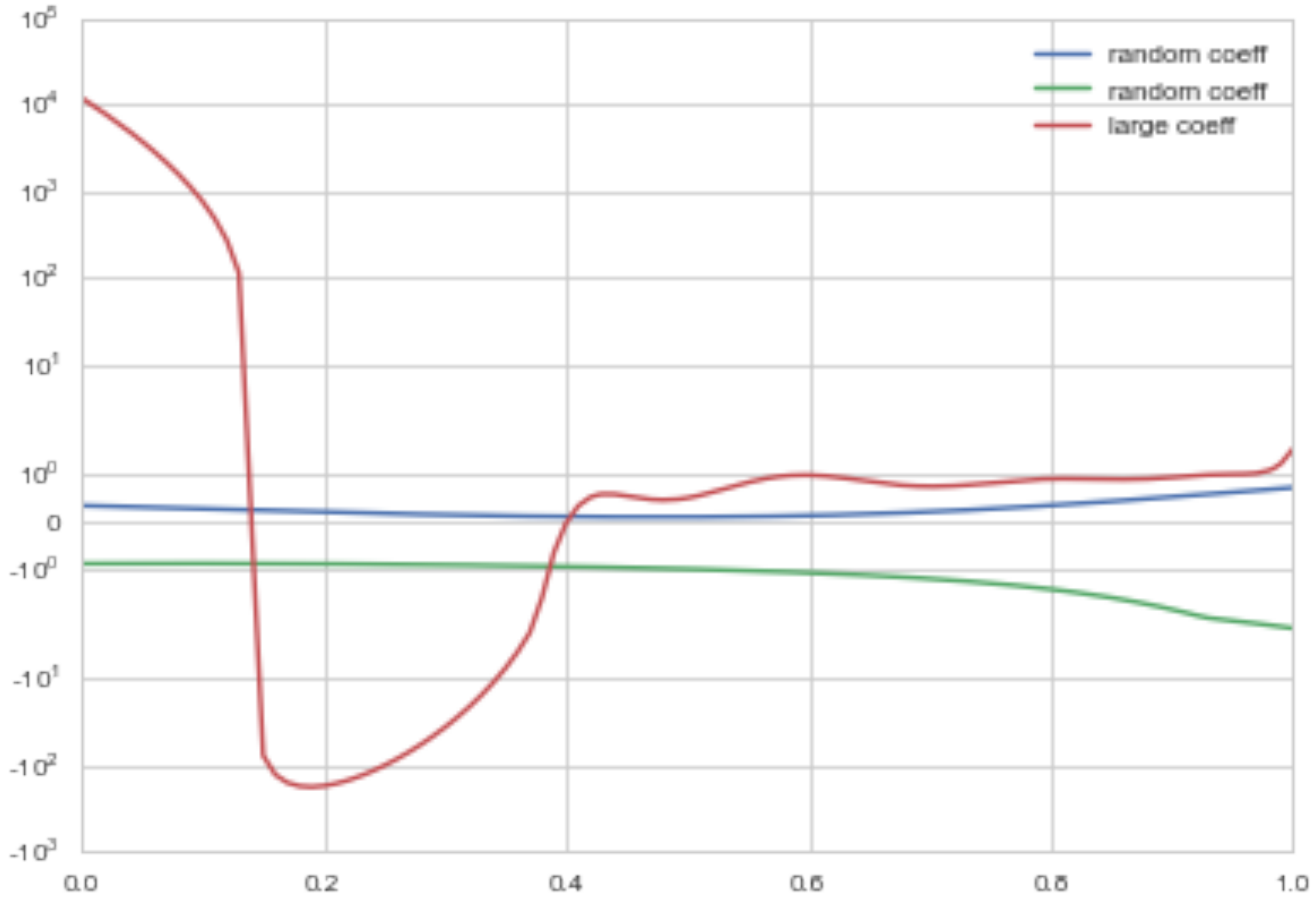
A polynomial looks so:

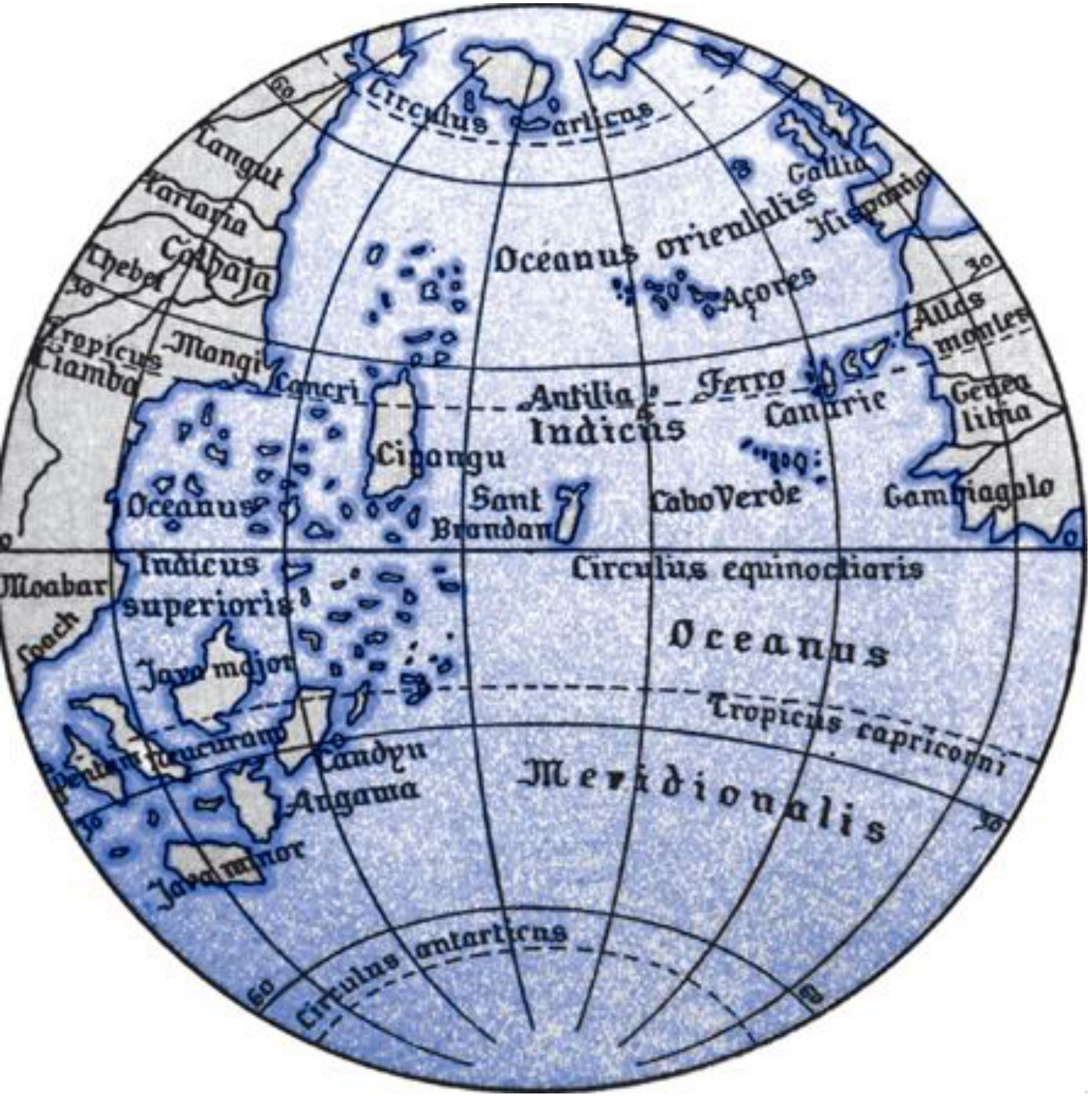
$$h(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2 + \ldots + \theta_n x^n = \sum_{i=0}^{n} \theta_i x^i$$

All polynomials of a degree or complexity d constitute a hypothesis space.

$$\mathcal{H}_1 : h_1(x) = \theta_0 + \theta_1 x$$

$$\mathcal{H}_{20} : h_{20}(x) = \sum_{i=0}^{20} \theta_i x^i$$





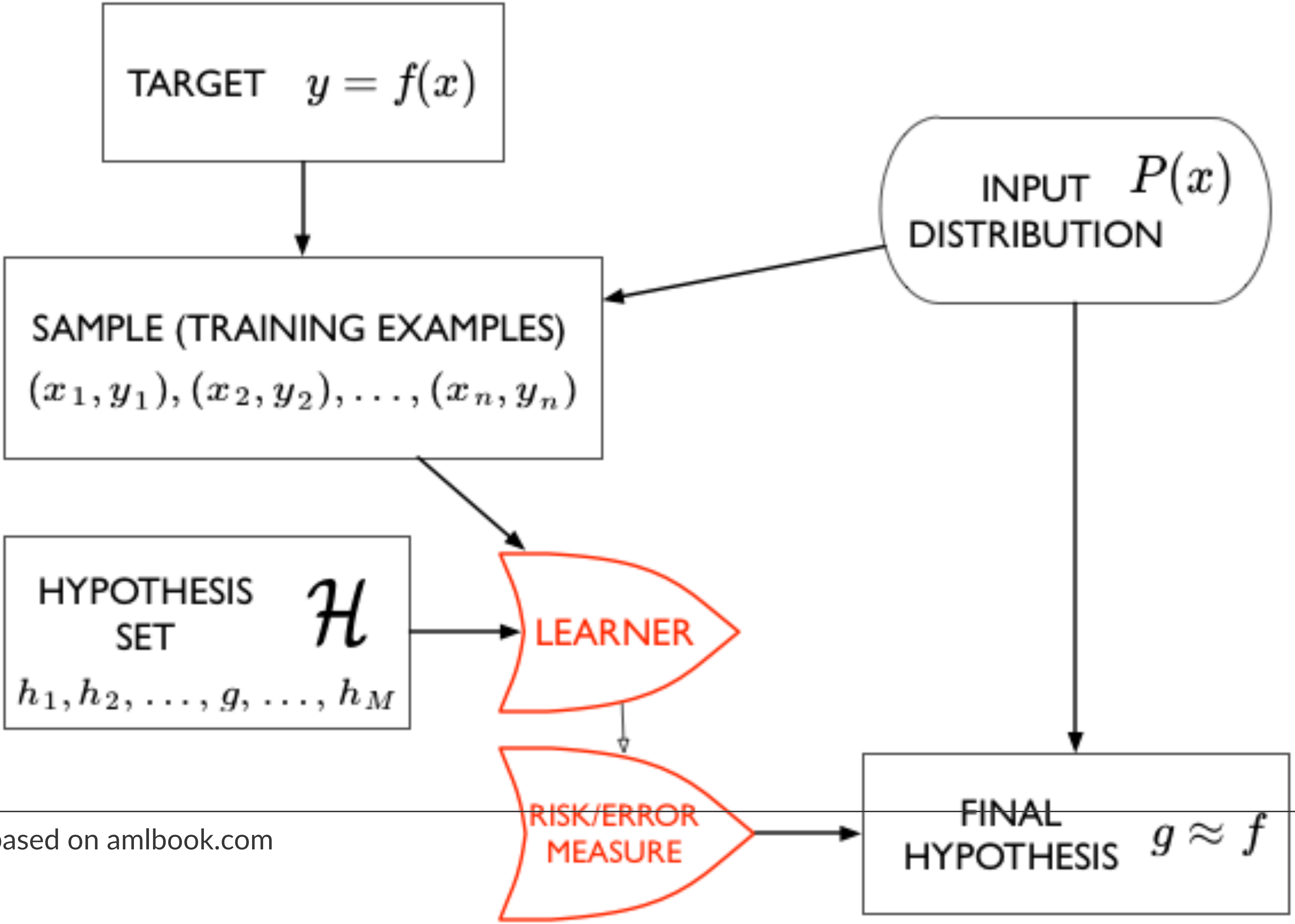
SMALL World vs BIG World

- *Small World* answers the question: given a model class (i.e. a Hypothesis space, whats the best model in it). It involves parameters. Its model checking.
- *BIG World* compares model spaces. Its model comparison with or without "hyperparameters".

Approximation

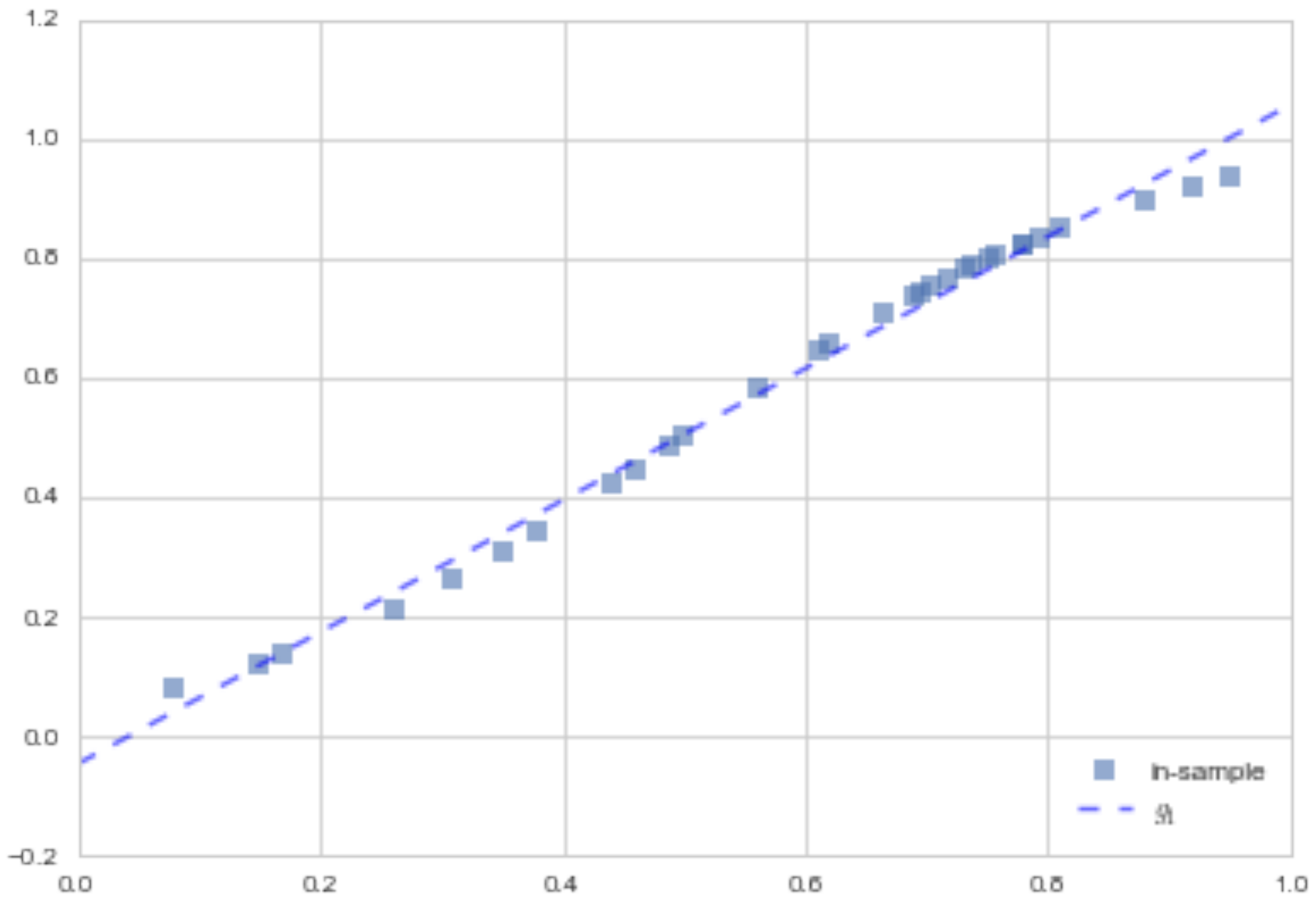
Learning Without Noise...

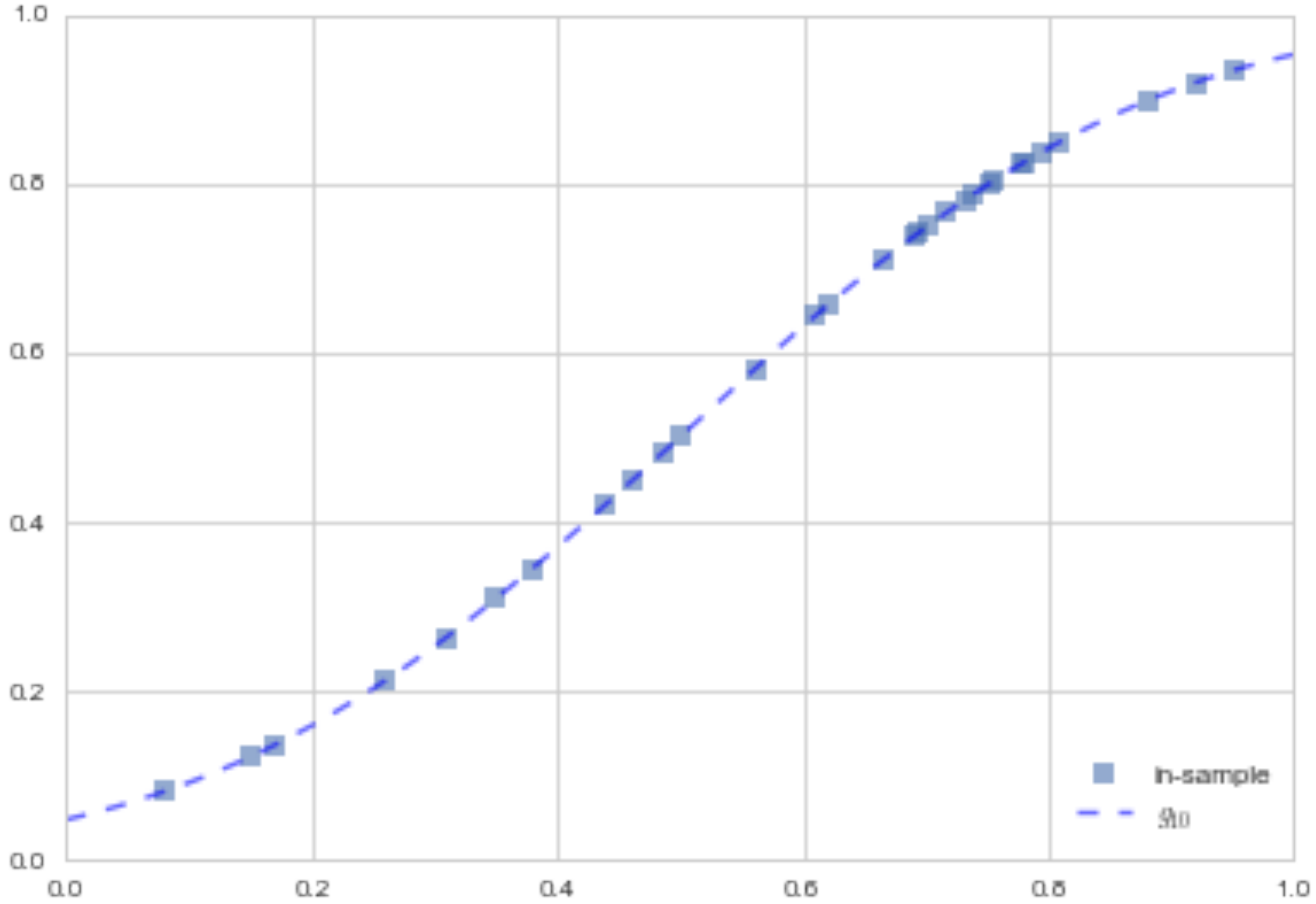
*



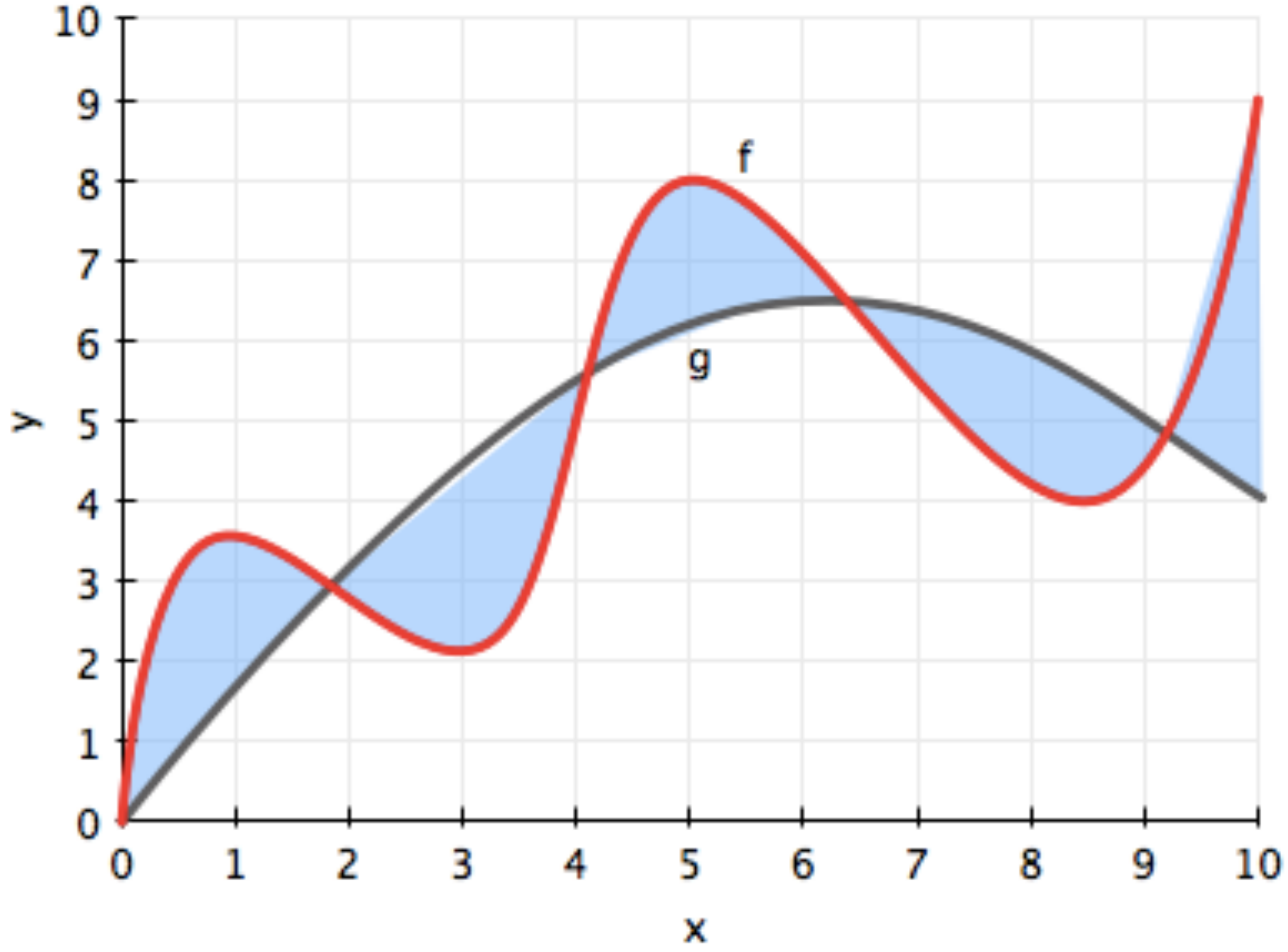
* image based on amlbook.com

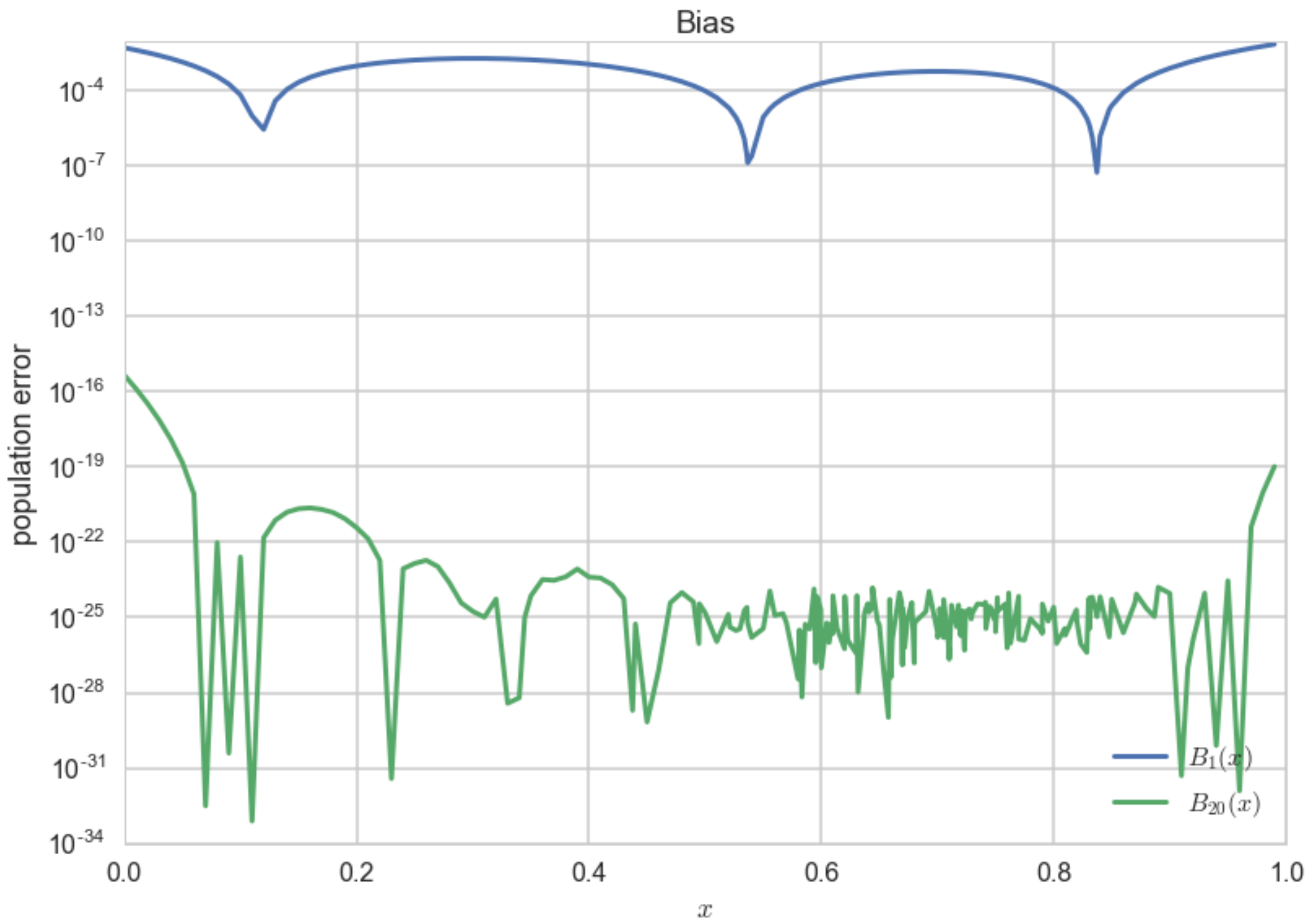
30 points of data. Which fit is better? Line in $\mathcal{H}_1$ or curve in $\mathcal{H}_{20}$?





Bias or Mis-specification Error





Sources of Variability

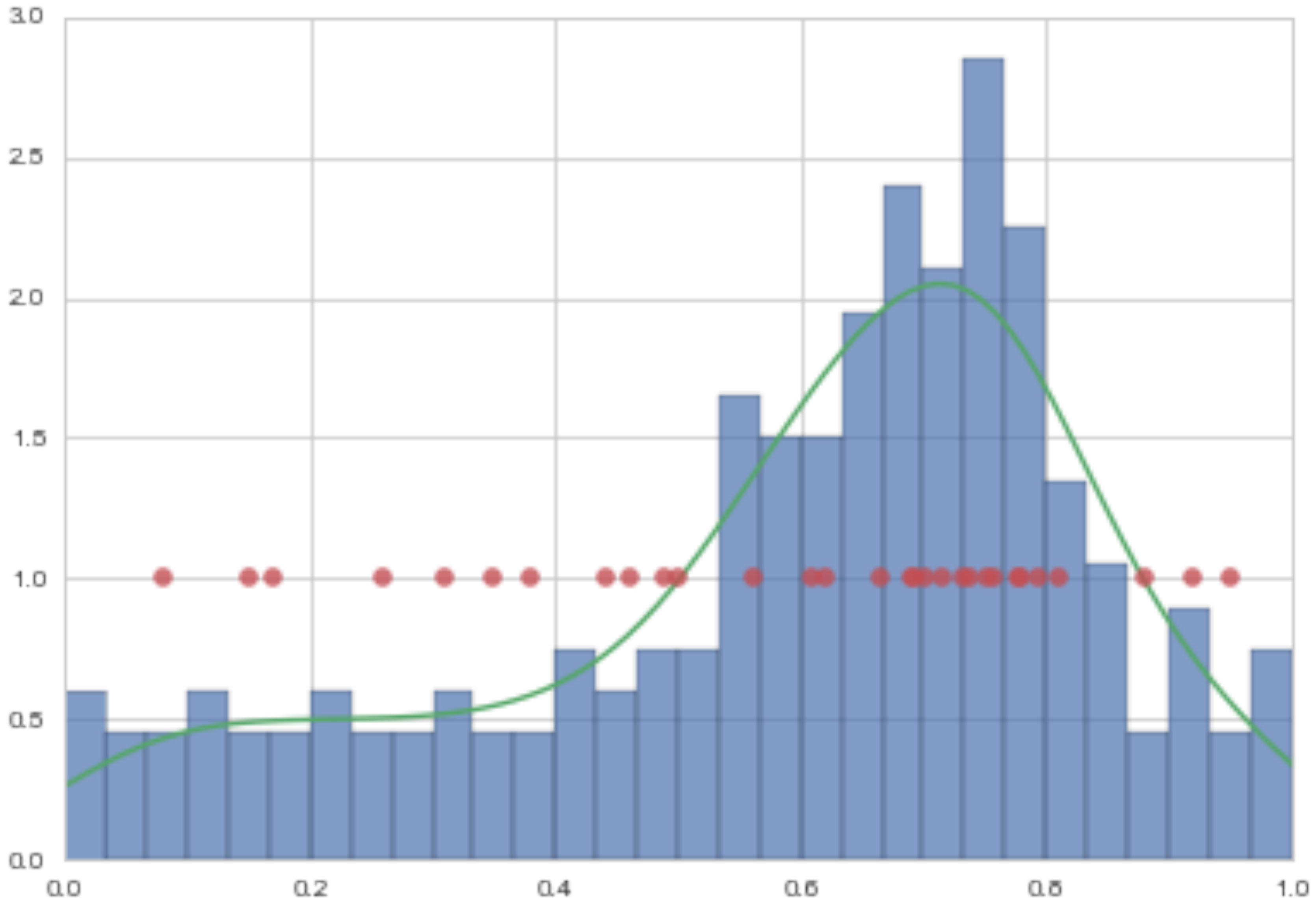
- sampling (induces variation in a mis-specified model)
- noise (the true $p(y|x)$)
- mis-specification

What is noise?

- noise comes from measurement error, missing features, etc
- sometimes it can be systematic as well, but its mostly random on account of being a combination of many small things...

THE REAL WORLD HAS NOISE

(or finite samples, usually both)



Statement of the Learning Problem

The sample must be representative of the population!

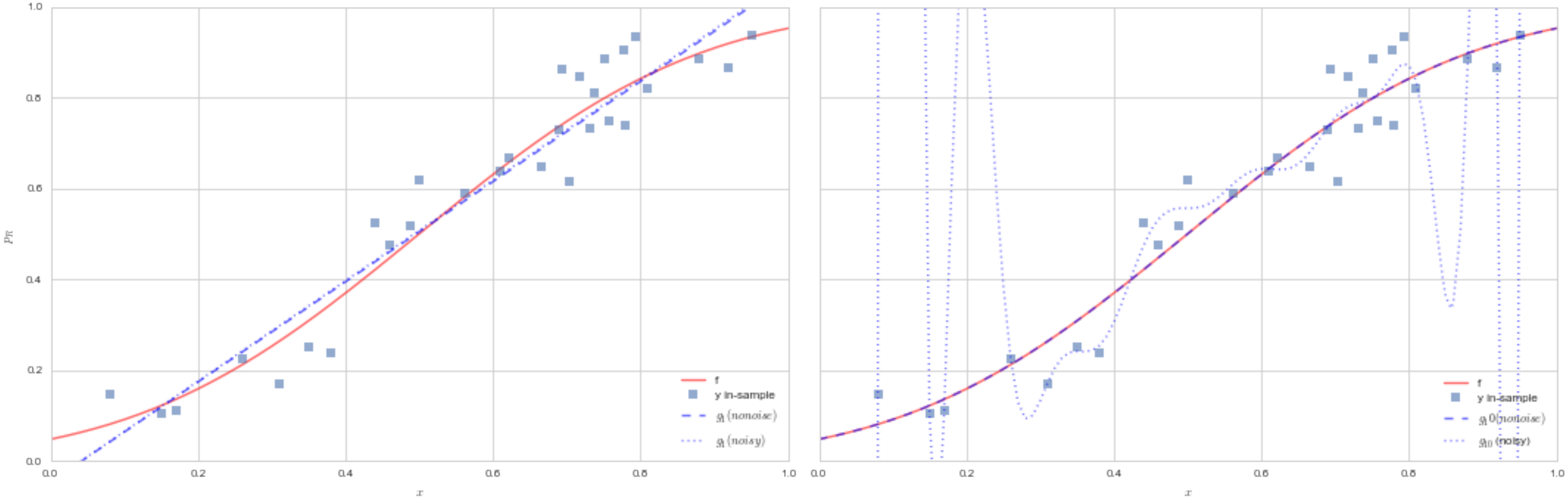
$$A : R_{\mathcal{D}}(g) \ \ smallest \, on \, \mathcal{H}$$
$$B : R_{out}(g) \approx R_{\mathcal{D}}(g)$$

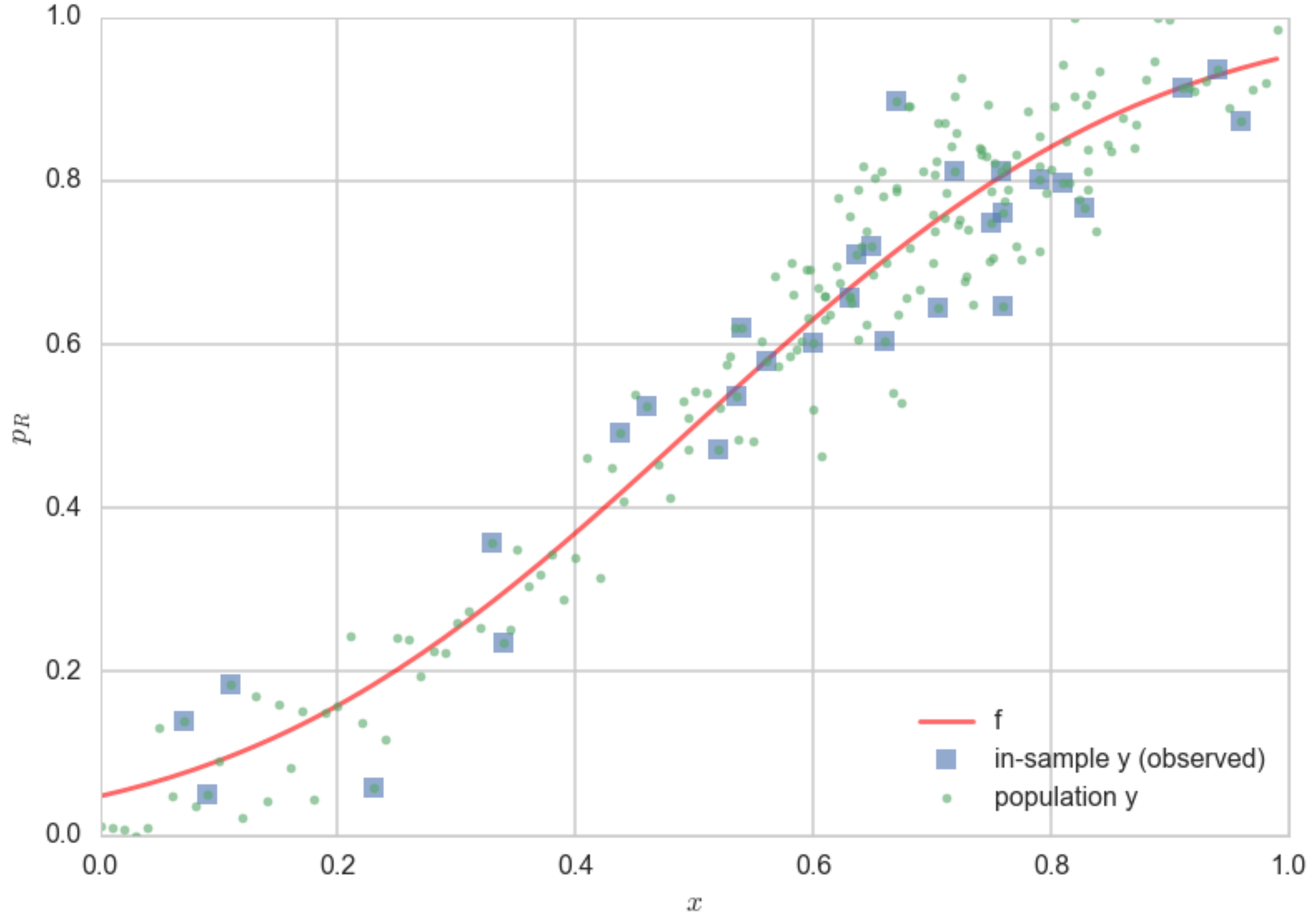
A: Empirical risk estimates in-sample risk.

B: Thus the out of sample risk is also small.

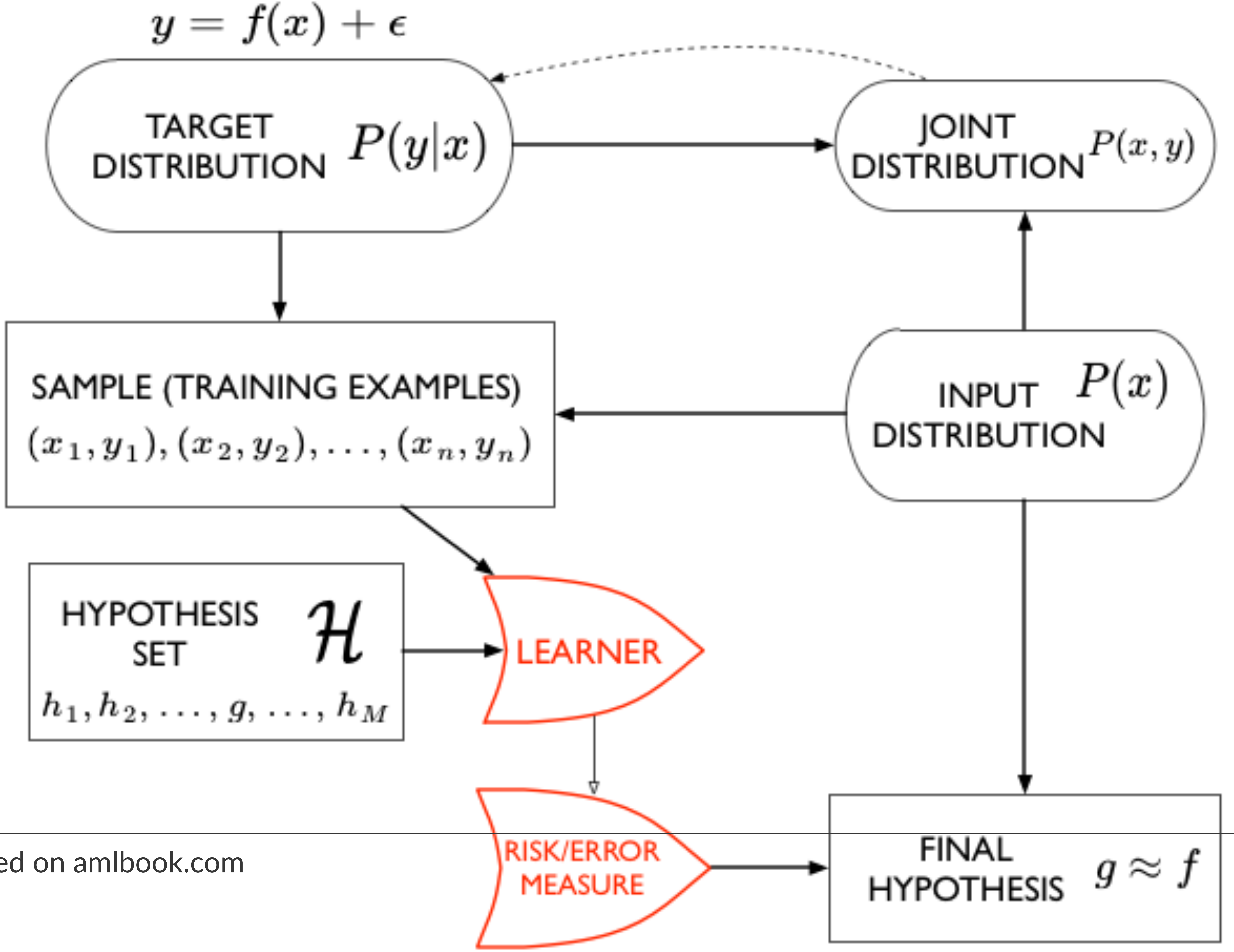
Which fit is better now?

The line or the curve?




1.0
0.8
0.6
0.4
0.2
0.0
p_R
0.0
0.2
0.4
0.6
0.8
1.0
x
f
in-sample y (observed)
population y

*

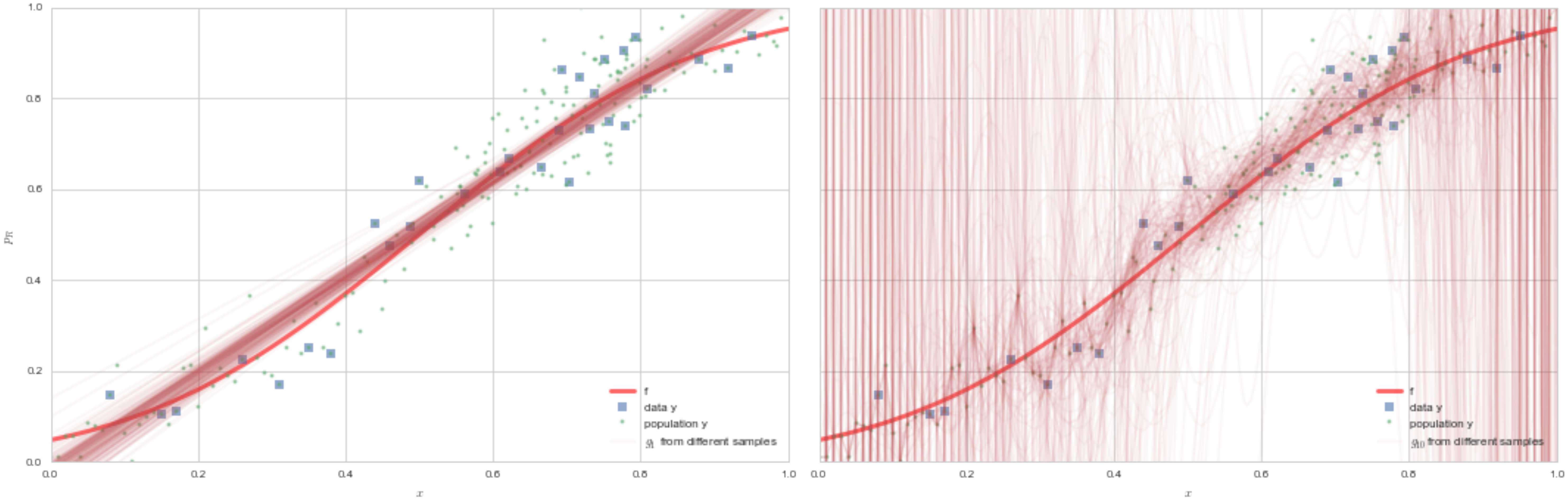


[*] image based on amlbook.com

Training sets

- look at fits on different "training sets $\mathcal{D}$"
- in other words, different samples
- in real life we are not so lucky, usually we get only one sample
- but lets pretend, shall we?

UNDERFITTING (Bias) vs OVERFITTING (Variance)



Risk for a given h

Define:

$$R_{out}(h) = E_{p(x,y)}[(h(x) - y)^2 | h] = \int dydx\, p(x,y)(h(x) - y)^2$$

$$R_{out}(h) = \int dx p(x,y)(h(x) - f(x) - \epsilon)^2.$$

(we assume 0 mean finite-variance noise ϵ)

Bayes Risk

$$R^* = \inf_h R_{out}(h) = \inf_h \int dx p(x,y)(h(x)-y)^2.$$

Its the minimum risk **ANY** model can achieve.

Want to get as close to it as possible.

Could infimum amongst all possible functions. OVERFITTING!

Instead restrict to a particular Hypothesis Set: $\mathcal{H}$.

Bayes Risk for Regression (population)

$$R_{out}(h) = \int dx p(x,y)(h(x)-y)^2.$$

$$= E_X E_{Y|X}[(h-y)^2] = E_X E_{Y|X}[(h-f+f-y)^2]$$

where f is chosen to be $r(x) = E_{Y|X}[y]$ is the "regression" function.

$$R_{out}(h) = E_X[(h-f)^2] + R^*; R^* = E_X E_{Y|X}[(f-y)^2] = \sigma^2$$

For 0 mean, finite variance, then, σ^2, the noise of ϵ, is the Bayes Risk, also called the irreducible error.

Empirical Risk Minimization

- Assume $(x_i, y_i) \sim P(x, y)$ (use empirical distrib)
- Minimize $\hat{R_{\mathcal{D}}} = \frac{1}{N} \sum_{i \in \mathcal{D}} L(y_i, h(x_i))$
- Thus fit hypothesis $h = g_{\mathcal{D}}$, where $\mathcal{D}$ is our training sample.
- $R_{out}(g_{\mathcal{D}})$ is now stochastic, so calculate:
- $\langle R \rangle = E_{\mathcal{D}}[R_{out}(g_{\mathcal{D}})]$

$$\langle R \rangle = E_{\mathcal{D}}[R_{out}(g_{\mathcal{D}})] = E_{\mathcal{D}} E_{p(x,y)}[(g_{\mathcal{D}}(x) - y)^2]$$

$\bar{g} = E_{\mathcal{D}}[g_{\mathcal{D}}] = (1/M) \sum_{\mathcal{D}} g_{\mathcal{D}}$. Then,

$$\langle R \rangle = E_{p(x)}[E_{\mathcal{D}}[(g_{\mathcal{D}} - \bar{g})^2]] + E_{p(x)}[(f - \bar{g})^2] + \sigma^2$$

This is the bias variance decomposition for regression. Or, written as $\langle R \rangle - R^*$, this is

`variance` + `bias`^{2}, or `estimation-error` + `approximation-error`

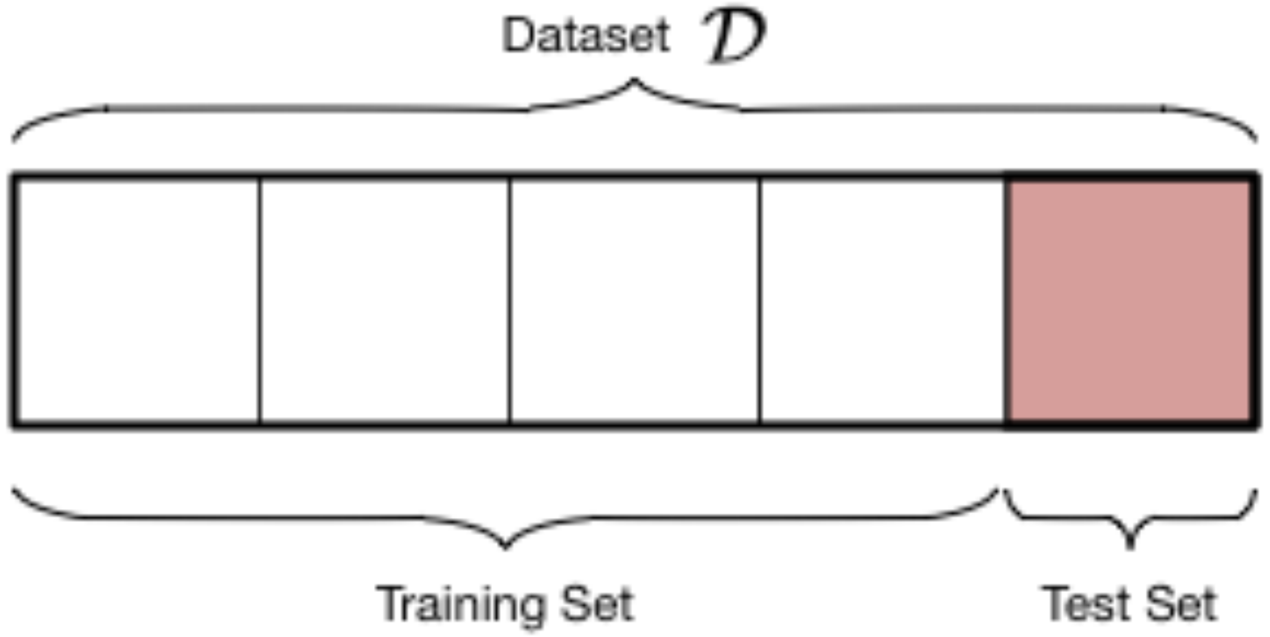
- first term is **variance**, squared error of the various fit g's from the average g, the hairiness.
- second term is **bias**, how far the average g is from the original f this data came from.
- third term is the **stochastic noise**, minimum error that this model will always have.

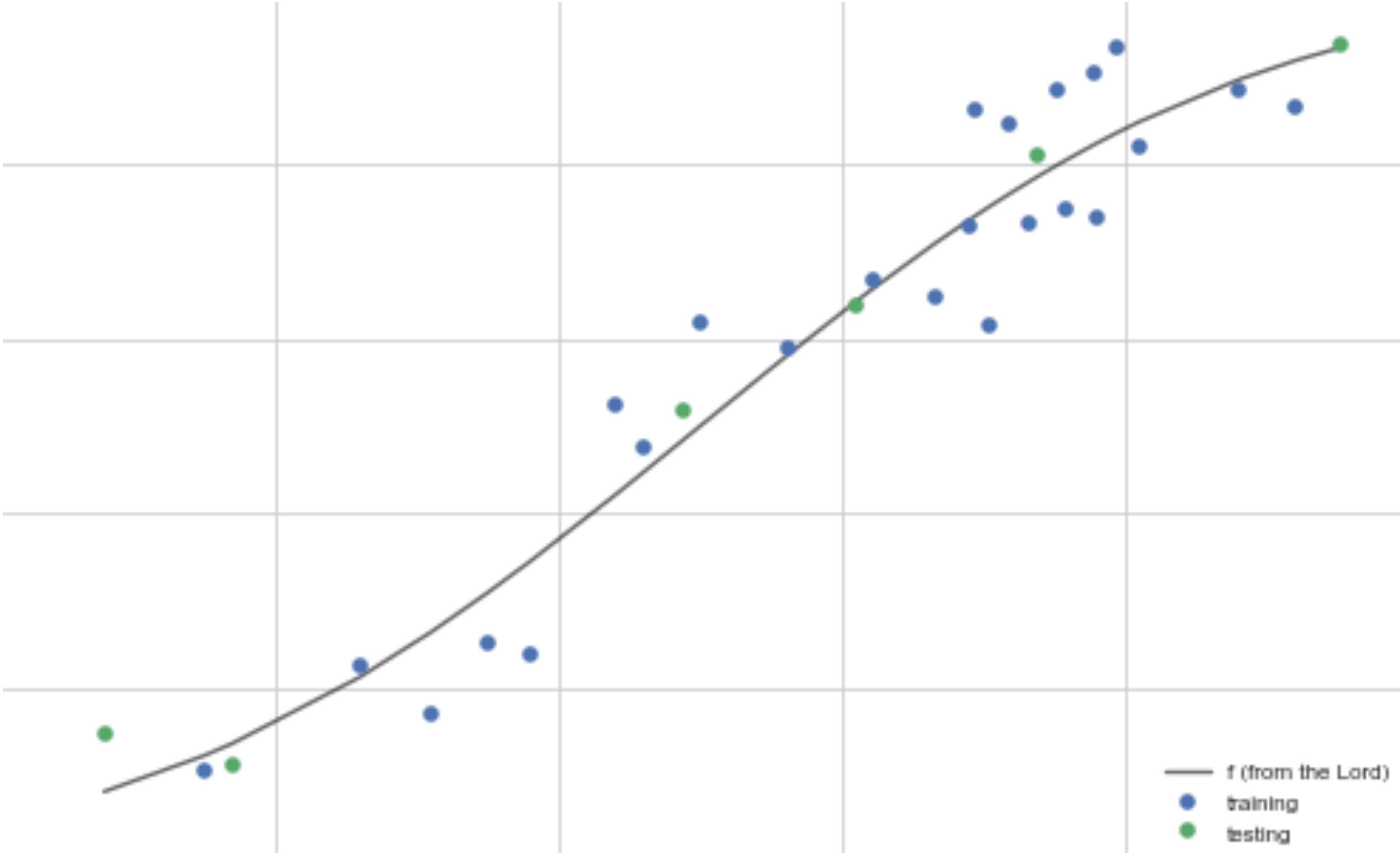
Goal of Learning: Build a function whose risk is closest to Bayes Risk (over a hypothesis set)

How do we estimate

out-of-sample or population error R_{out}

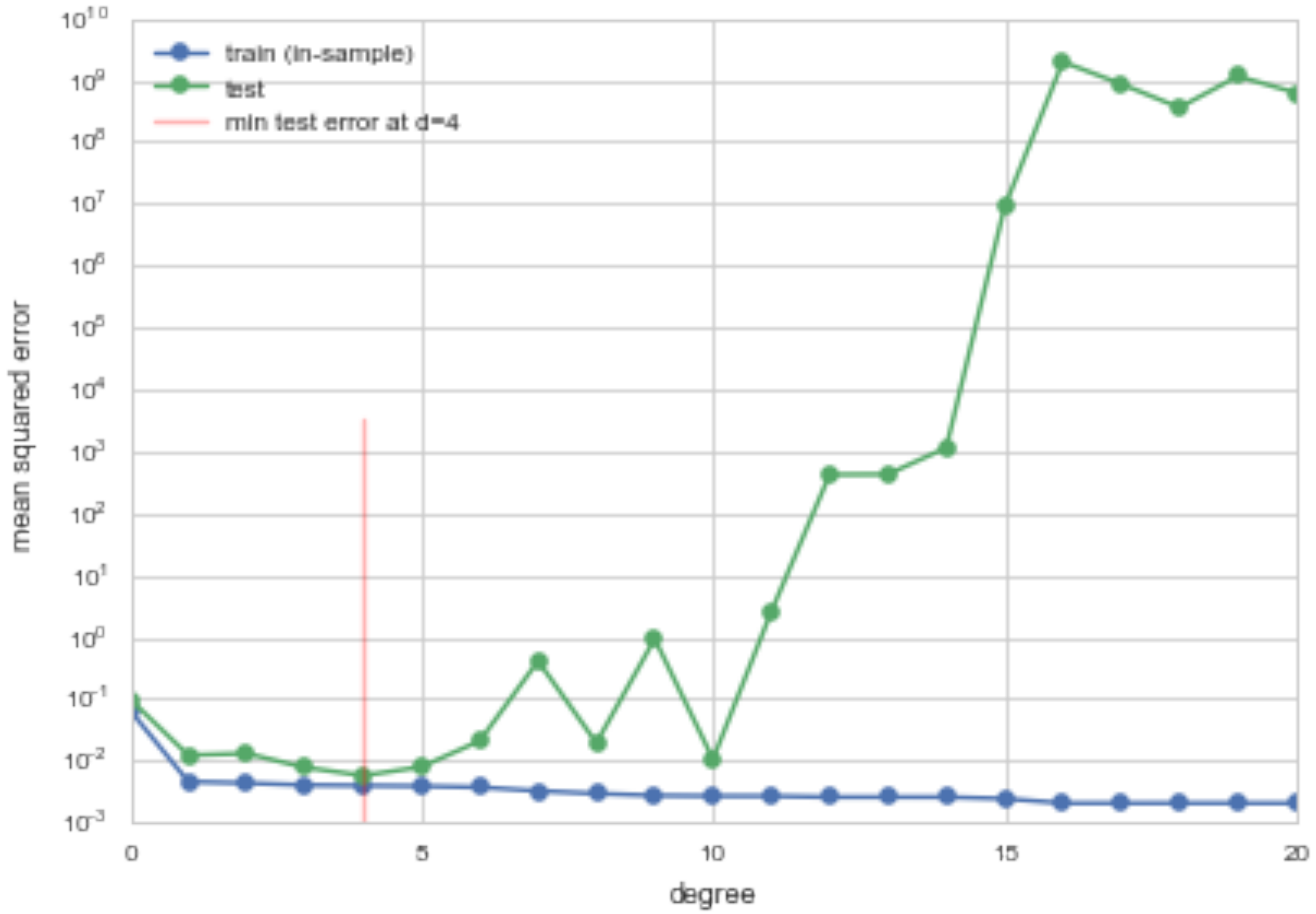
TRAIN AND TEST



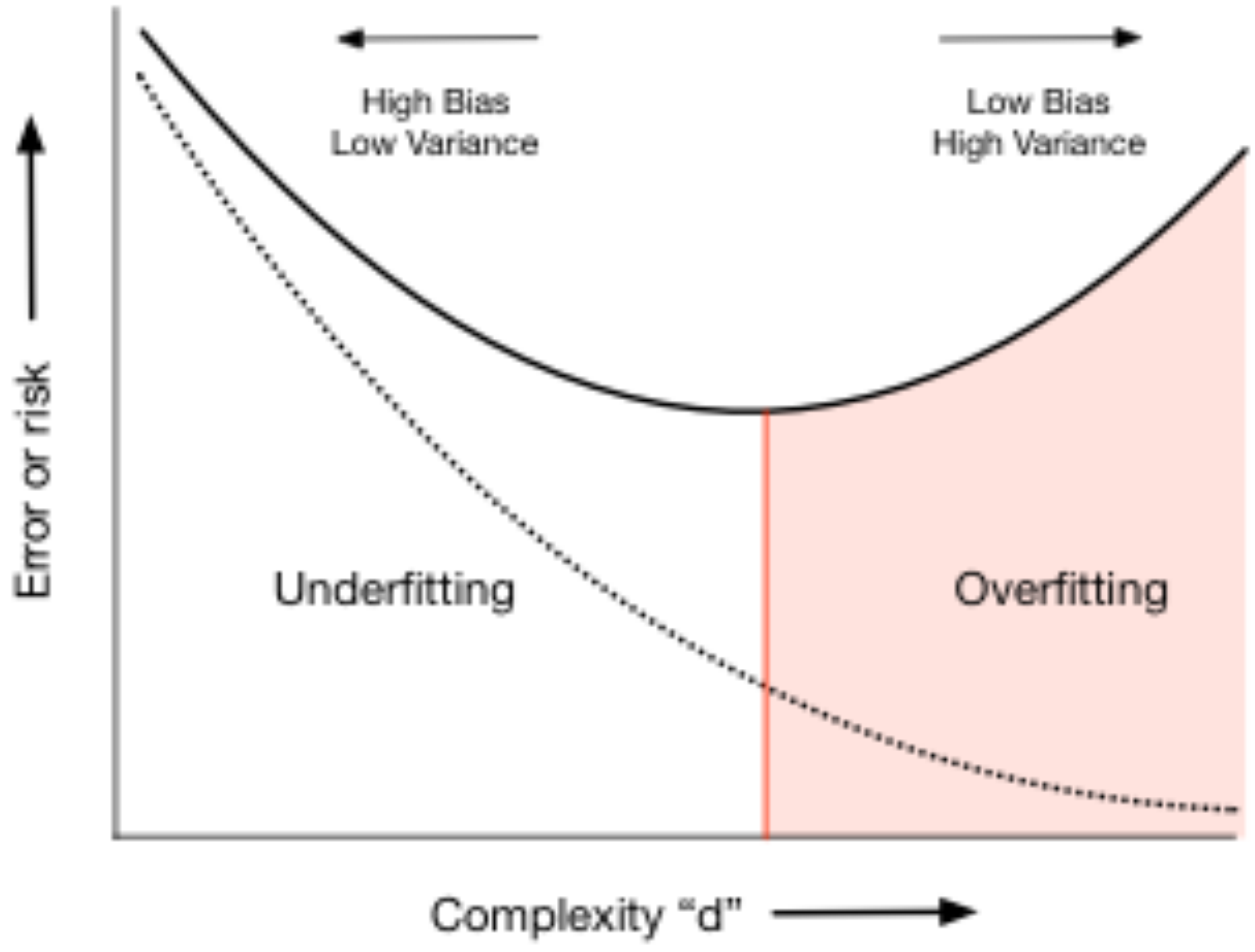


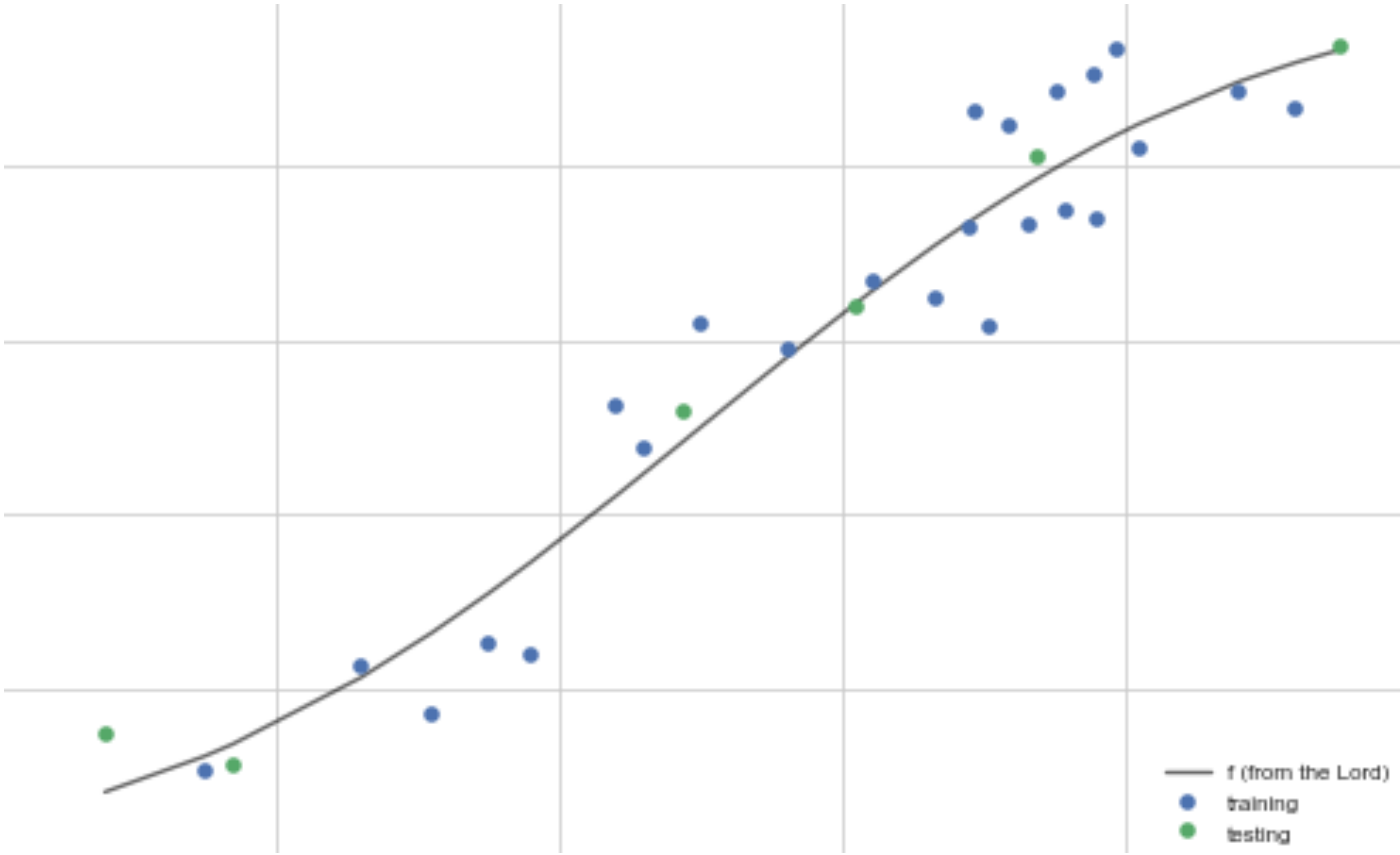
MODEL COMPARISON: A Large World approach

- want to choose which Hypothesis set is best
- it should be the one that minimizes risk
- but minimizing the training risk tells us nothing: interpolation
- we need to minimize the training risk but not at the cost of generalization
- thus only minimize till test set risk starts going up

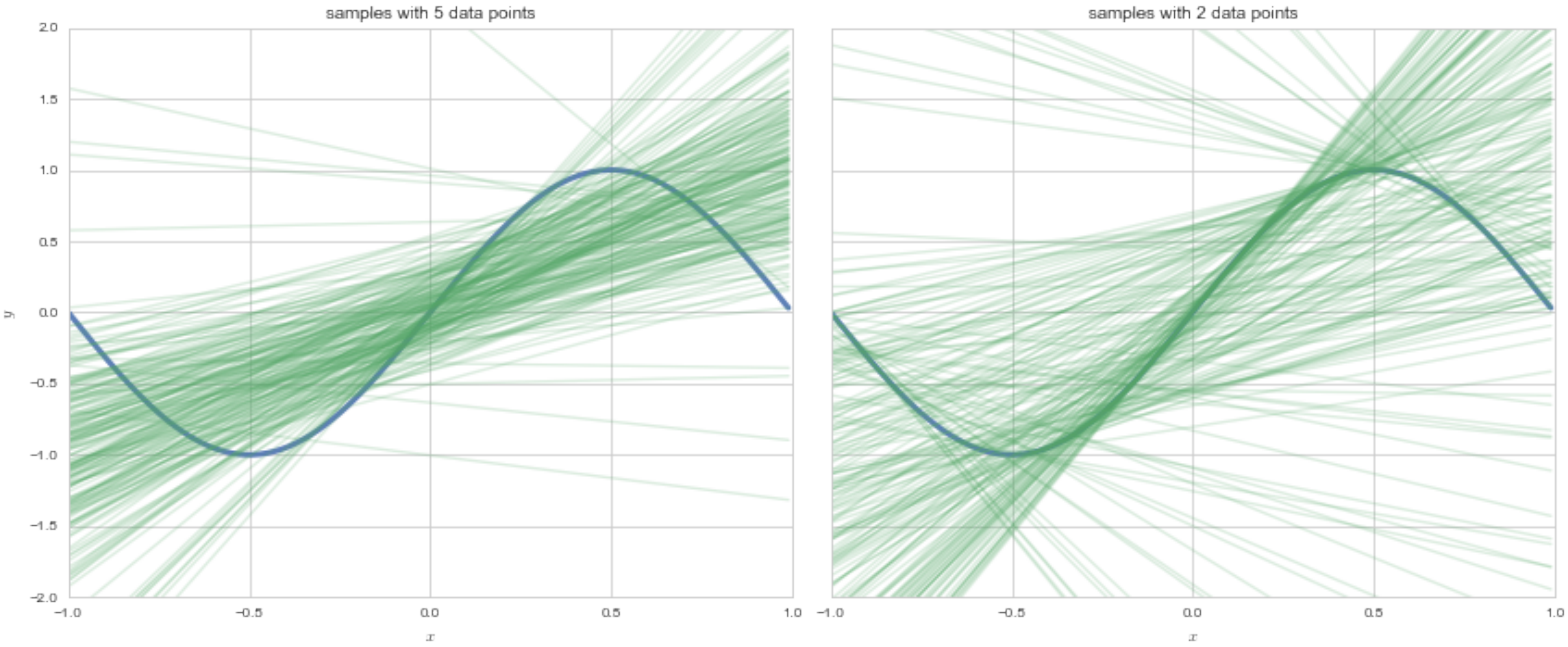


Complexity Plot




f (from the Lord)
training
testing

DATA SIZE MATTERS: straight line fits to a sine curve



Corollary: Must fit simpler models to less data! This will motivate the analysis of learning curves later.

Do we still have a test set?

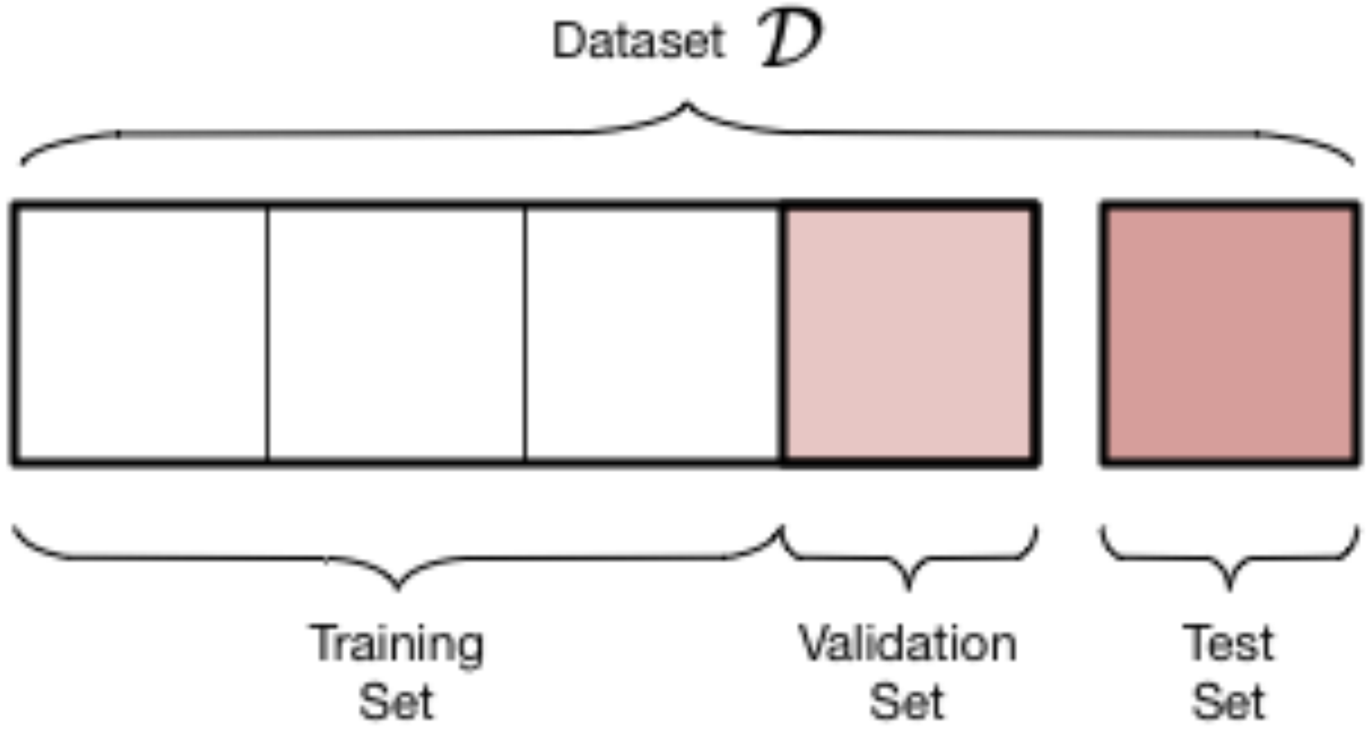
Trouble:

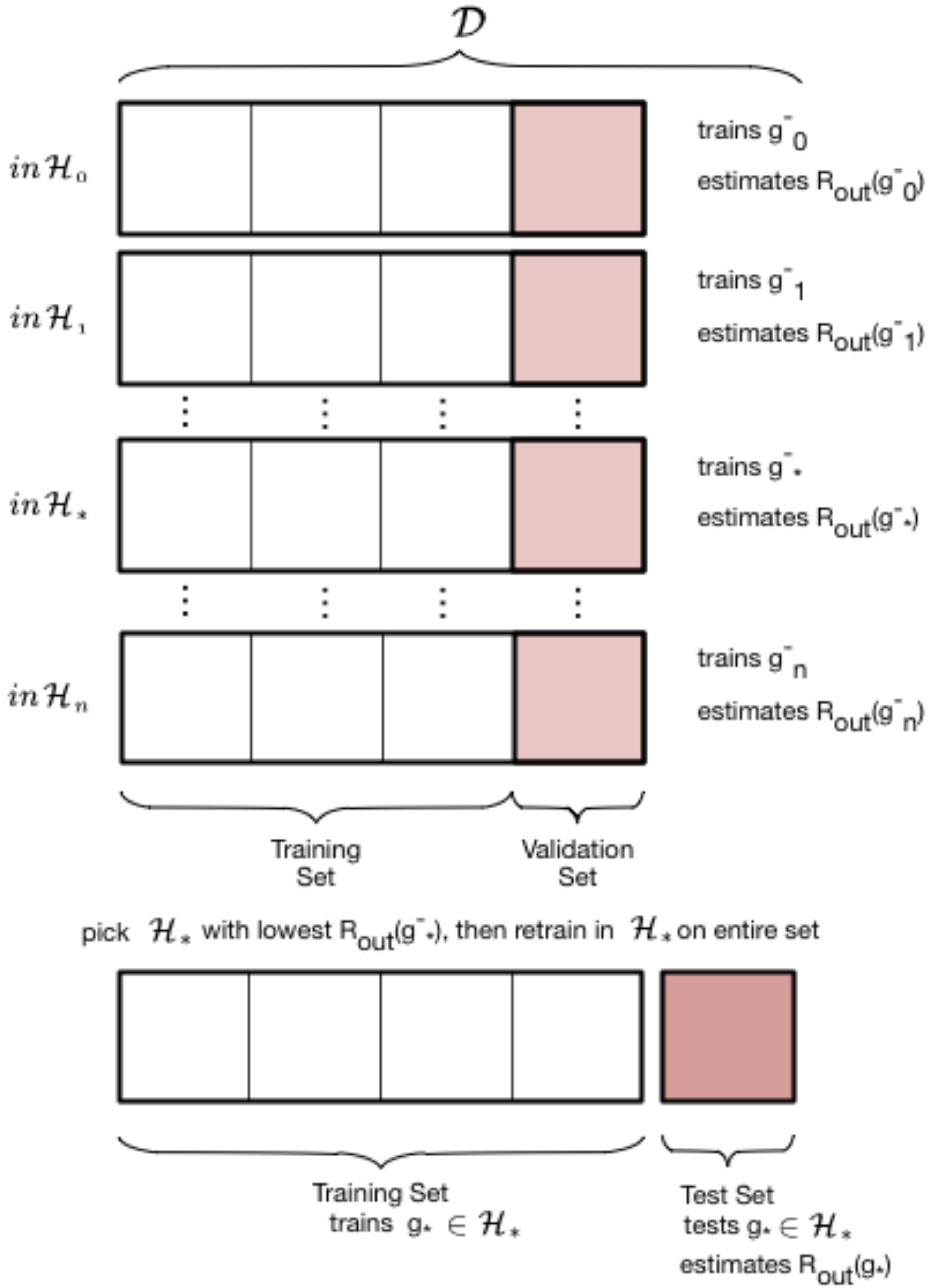
- no discussion on the error bars on our error estimates
- "visually fitting" a value of $d \implies$ contaminated test set.

The moment we **use it in the learning process, it is not a test set**.

VALIDATION

- train-test not enough as we *fit* for d on test set and contaminate it
- thus do train-validate-test





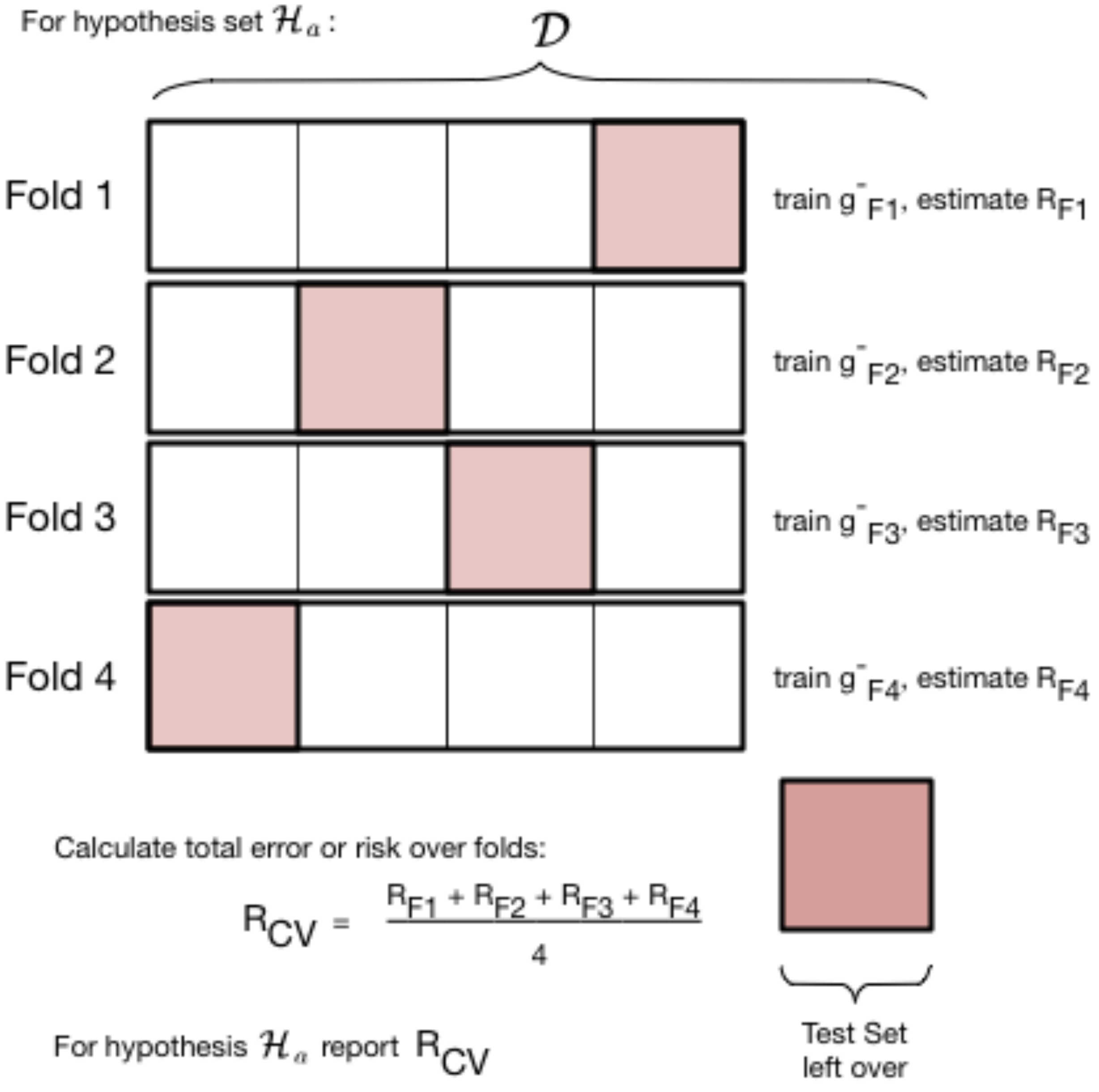
usually we want to fit a hyperparameter

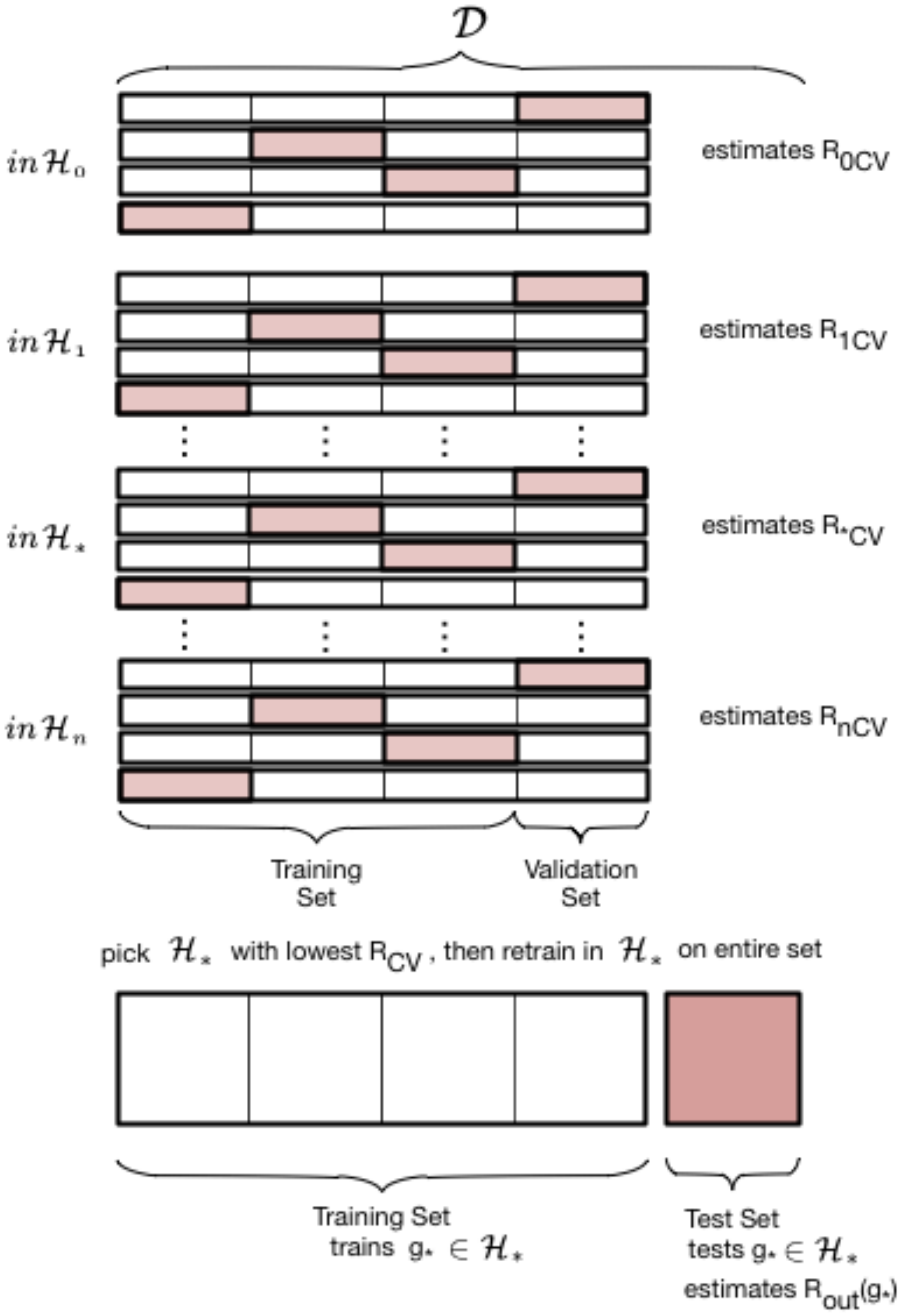
- we **wrongly** already attempted to fit d on our previous test set.
- choose the d, g^{-*} combination with the lowest validation set risk.
- $R_{val}(g^{-*}, d^*)$ has an optimistic bias since d effectively fit on validation set

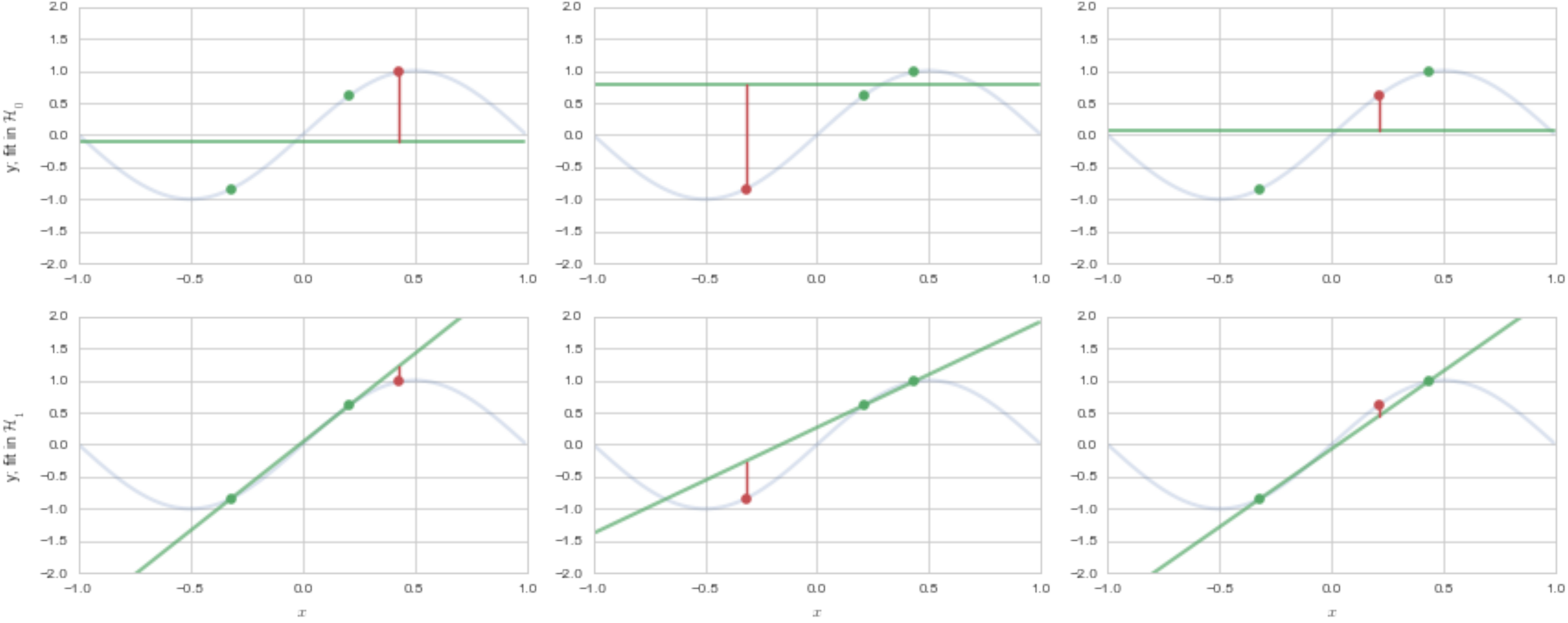
Then Retrain on entire set!

- finally retrain on the entire train+validation set using the appropriate d^*
- works as training for a given hypothesis space with more data typically reduces the risk even further.

CROSS-VALIDATION







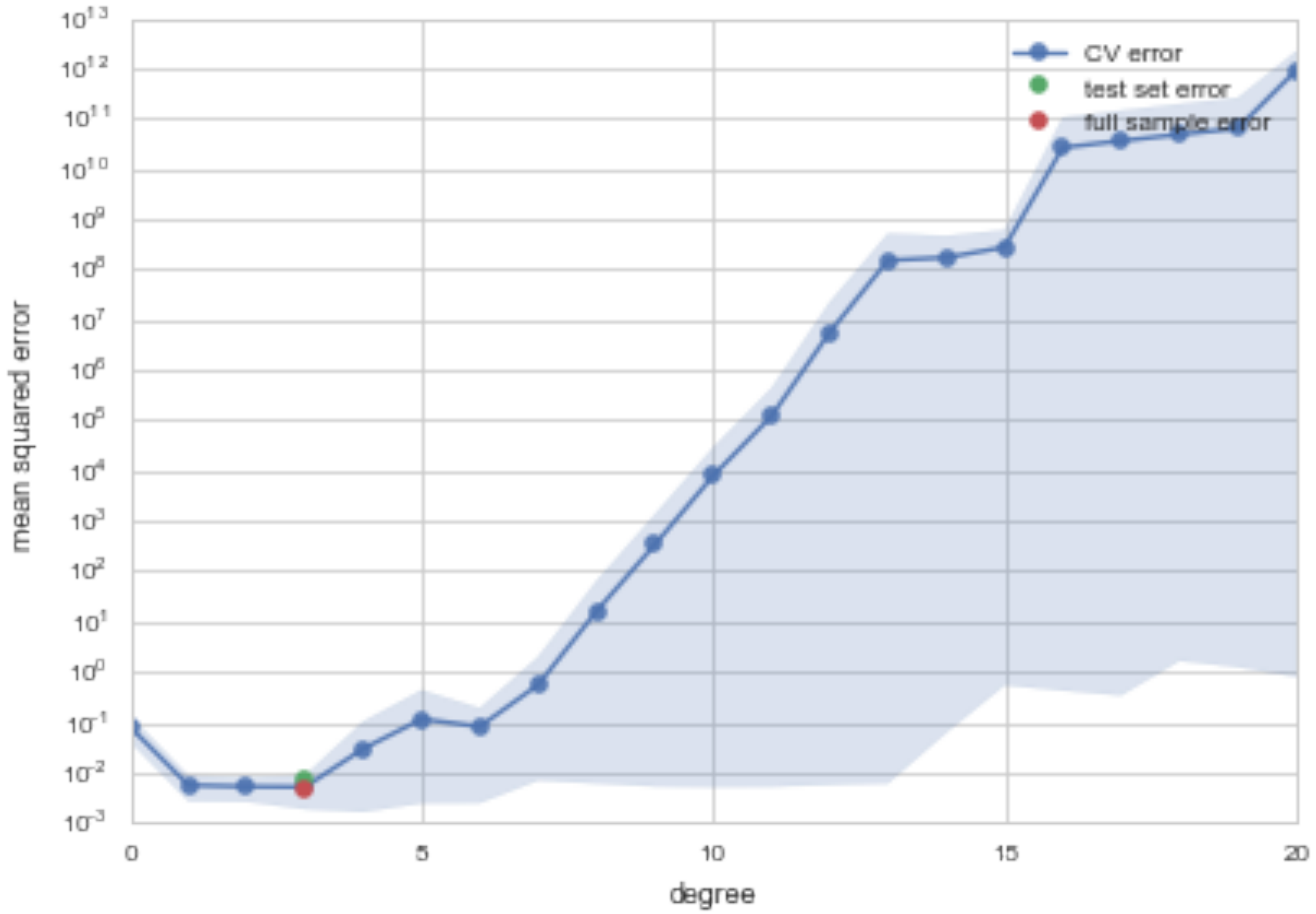
y; fit in $\mathcal{H}_0$
y; fit in $\mathcal{H}_1$
x
x
x

CROSS-VALIDATION

is

- a resampling method
- robust to outlier validation set
- allows for larger training sets
- allows for error estimates

Here we find $d = 3$.



Cross Validation considerations

- validation process as one that estimates R_{out} directly, on the validation set. It's critical use is in the model selection process.
- once you do that you can estimate R_{out} using the test set as usual, but now you have also got the benefit of a robust average and error bars.
- key subtlety: in the risk averaging process, you are actually averaging over different g^- models, with different parameters.

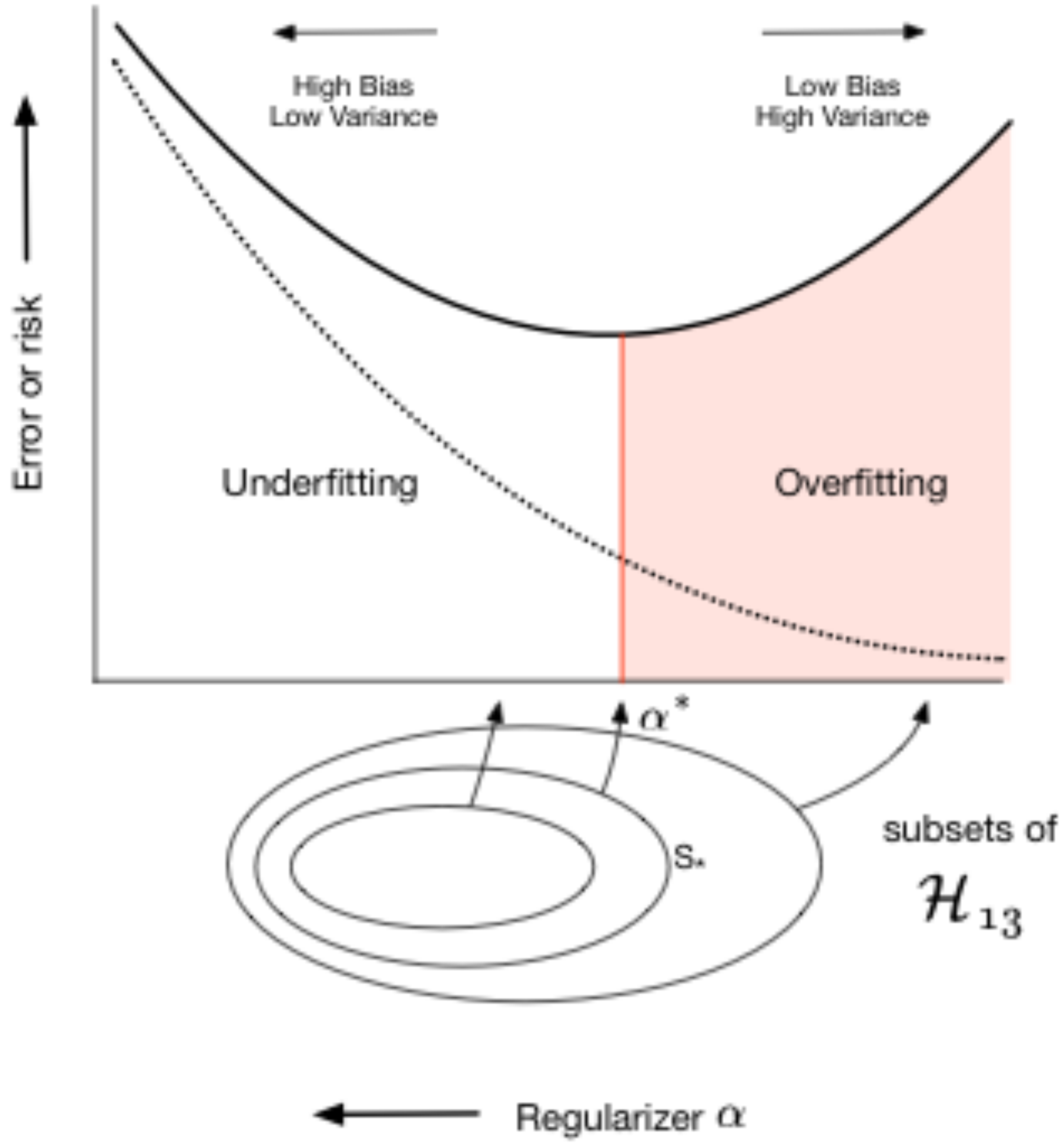
REGULARIZATION: A SMALL WORLD APPROACH

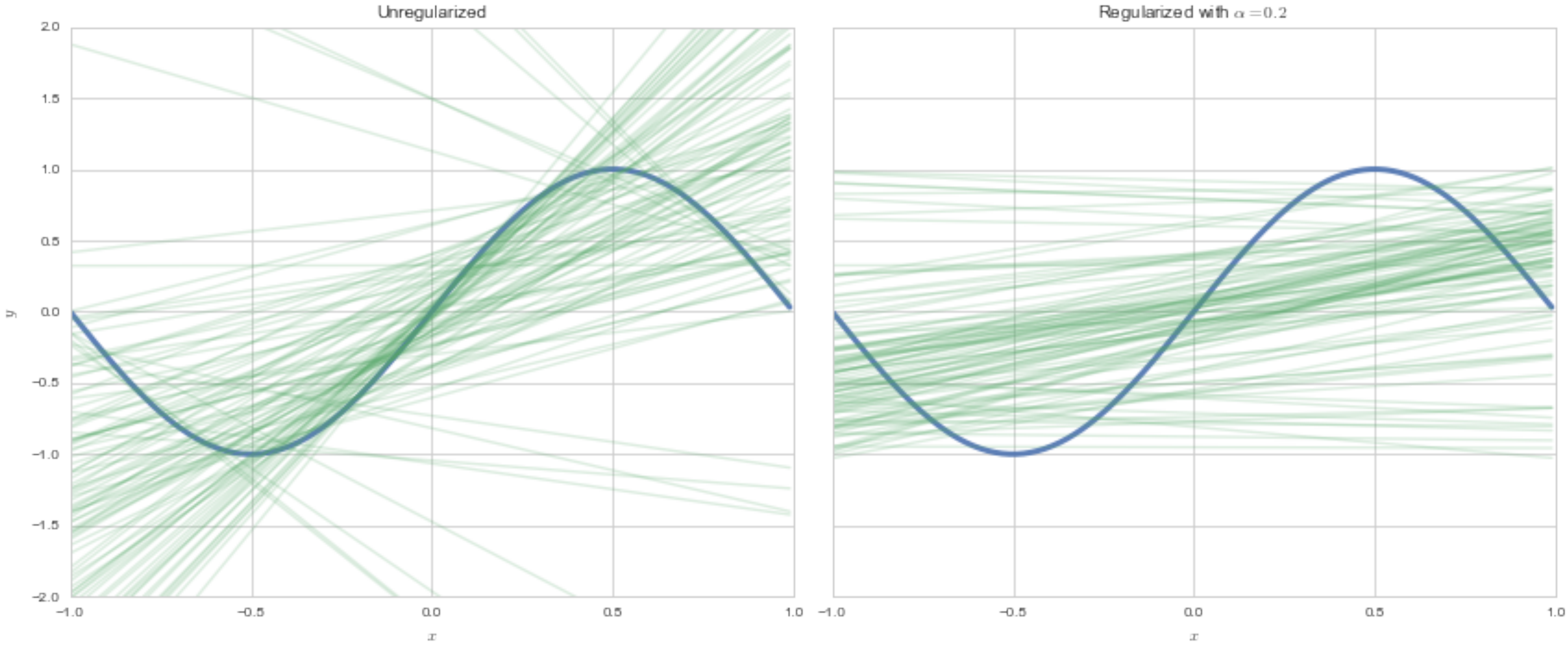
Keep higher a-priori complexity and impose a

complexity penalty

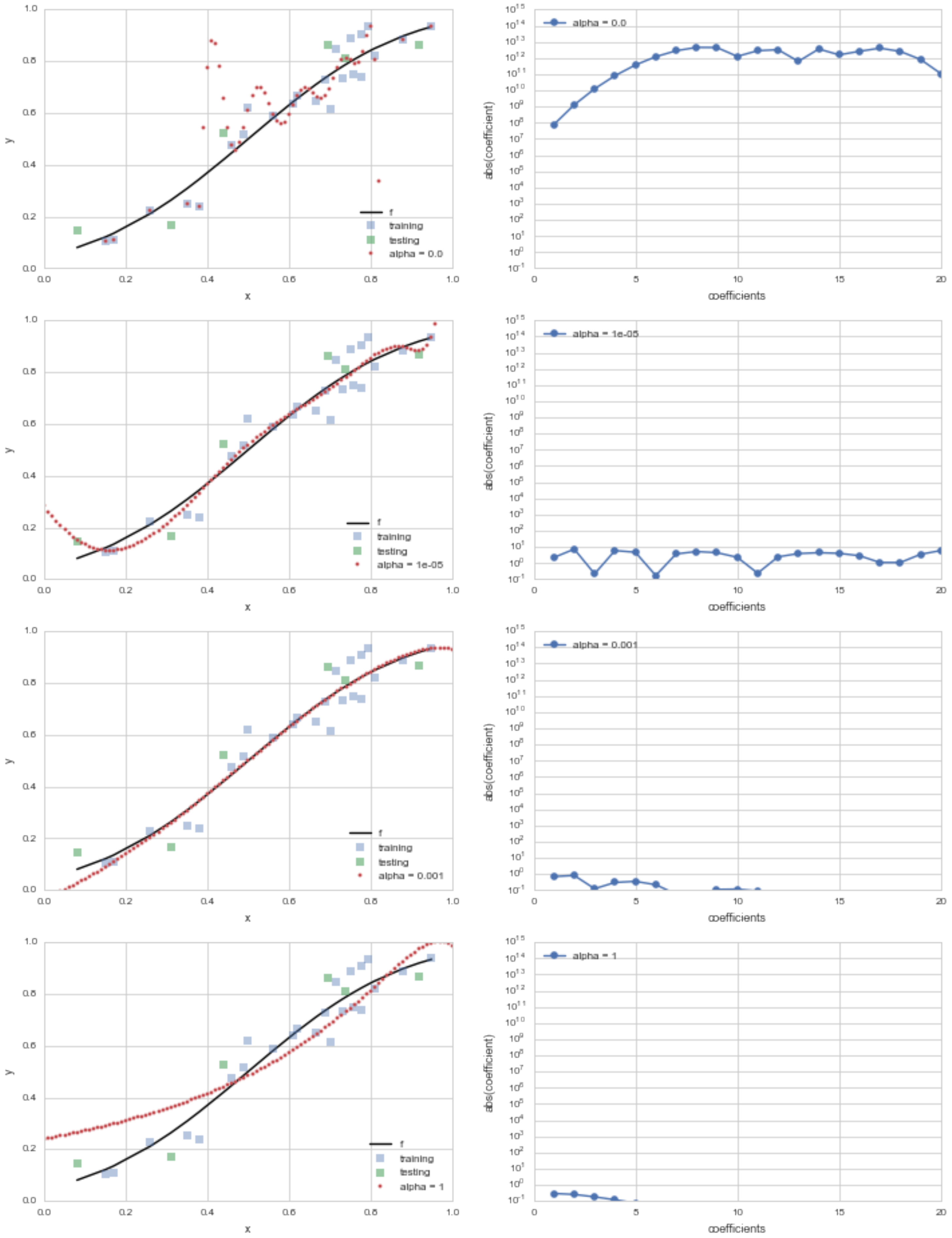
on risk instead, to choose a SUBSET of $\mathcal{H}_{big}$. We'll make the coefficients small:

$$\sum_{i=0}^{j} \theta_i^2 < C.$$





Unregularized
Regularized with $\alpha = 0.2$
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
y
−1.0
−0.5
0.0
0.5
1.0
x



REGULARIZATION

$$\mathcal{R}(h_j) = \sum_{y_i \in \mathcal{D}} (y_i - h_j(x_i))^2 + \alpha \sum_{i=0}^{j} \theta_i^2 .$$

As we increase α, coefficients go towards 0.

Lasso uses $\alpha \sum_{i=0}^{j} |\theta_i|$, sets coefficients to exactly 0.

Regularization with Cross-Validation