Lecture 5

Machine Learning

Office Hours

Always in B125, Maxwell Dworkin.

- anytime by appointment with any of us.
- Rahul: Tue and Thu 1.30pm to 2.30pm. ONLINE Thu 2.30-3.30pm.
- Will Tuesday 4 - 5 pm
- Wed Patrick and TBD 4 - 7:30

- 6:30 - 7:30 pm Wednesday will be online
- Thu Patrick 4-5:30, Peter 5:30 - 6:30

Last Times:

- Expectations, sample average
- The Law of large numbers and Monte Carlo
- Sampling Methods

Law of Large numbers (LLN)

- Expectations become sample averages. Convergence for large N.

$$E_f[g] = \int g(x)dF = \int g(x)f(x)dx$$
$$= \lim_{n \to \infty} \frac{1}{N} \sum_{x_i \sim f} g(x_i)$$

- for finite N a sample average
- thus expectations in the replication "dimension" come into play
- mean of sample means and standard error
- this is the sampling distribution
- CLT and all that jazz

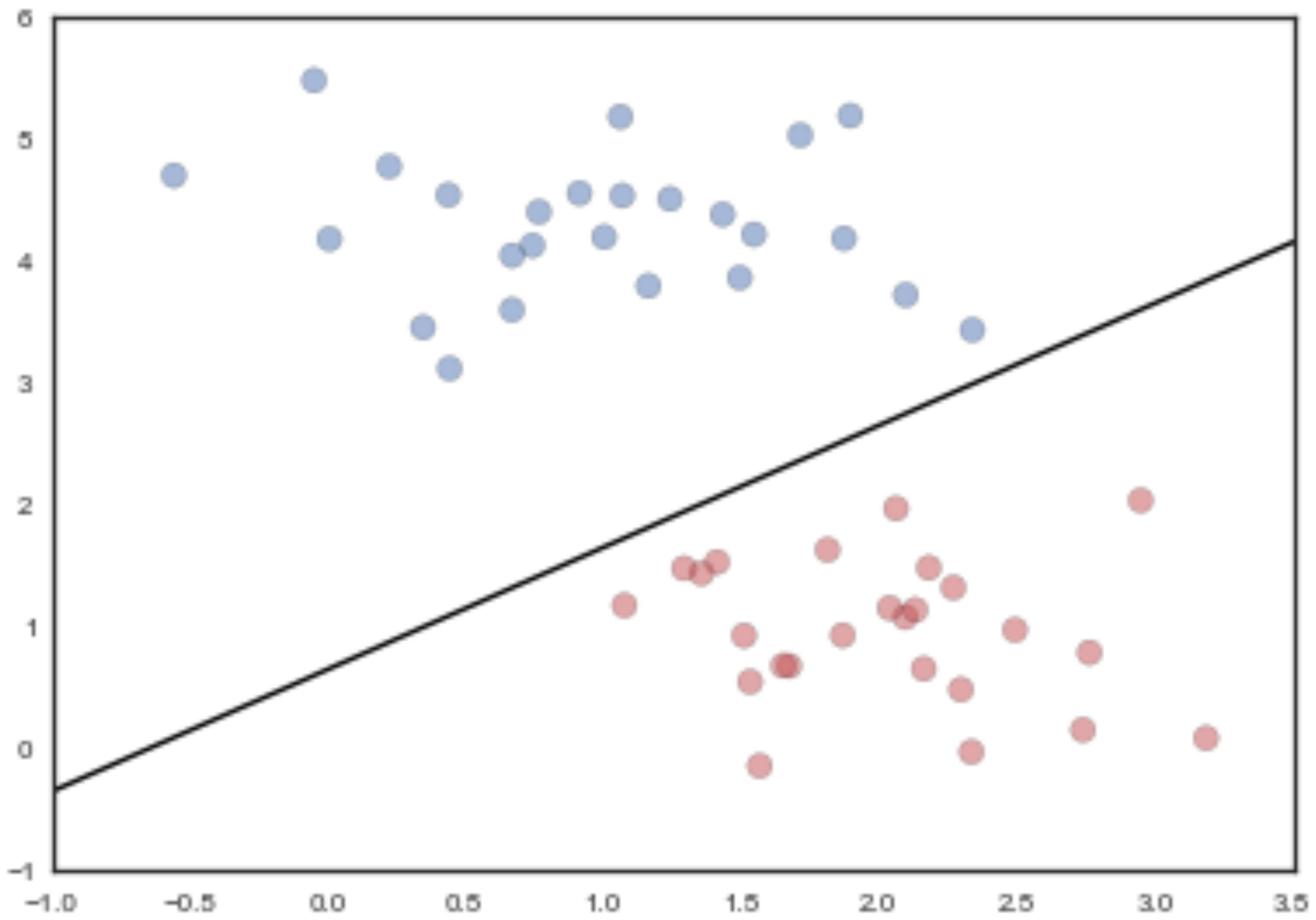
Today: machine Learning

- noiseless models, the approximation problems
- models with noise
- test sets and learning theory
- validation and cross-validation
- regularization

Why study this?

- isnt this a course in Stoch Opt and Bayes?
- application of law of large numbers
- establishes ideas of supervised learning
- learn validation for model selection
- bayes critical to understand machine learning

CLASSIFICATION



- will a customer churn?
- is this a check? For how much?
- a man or a woman?
- will this customer buy?
- do you have cancer?
- is this spam?

[j] image from code in http://bit.ly/1Azg29G

MLE for Logistic Regression

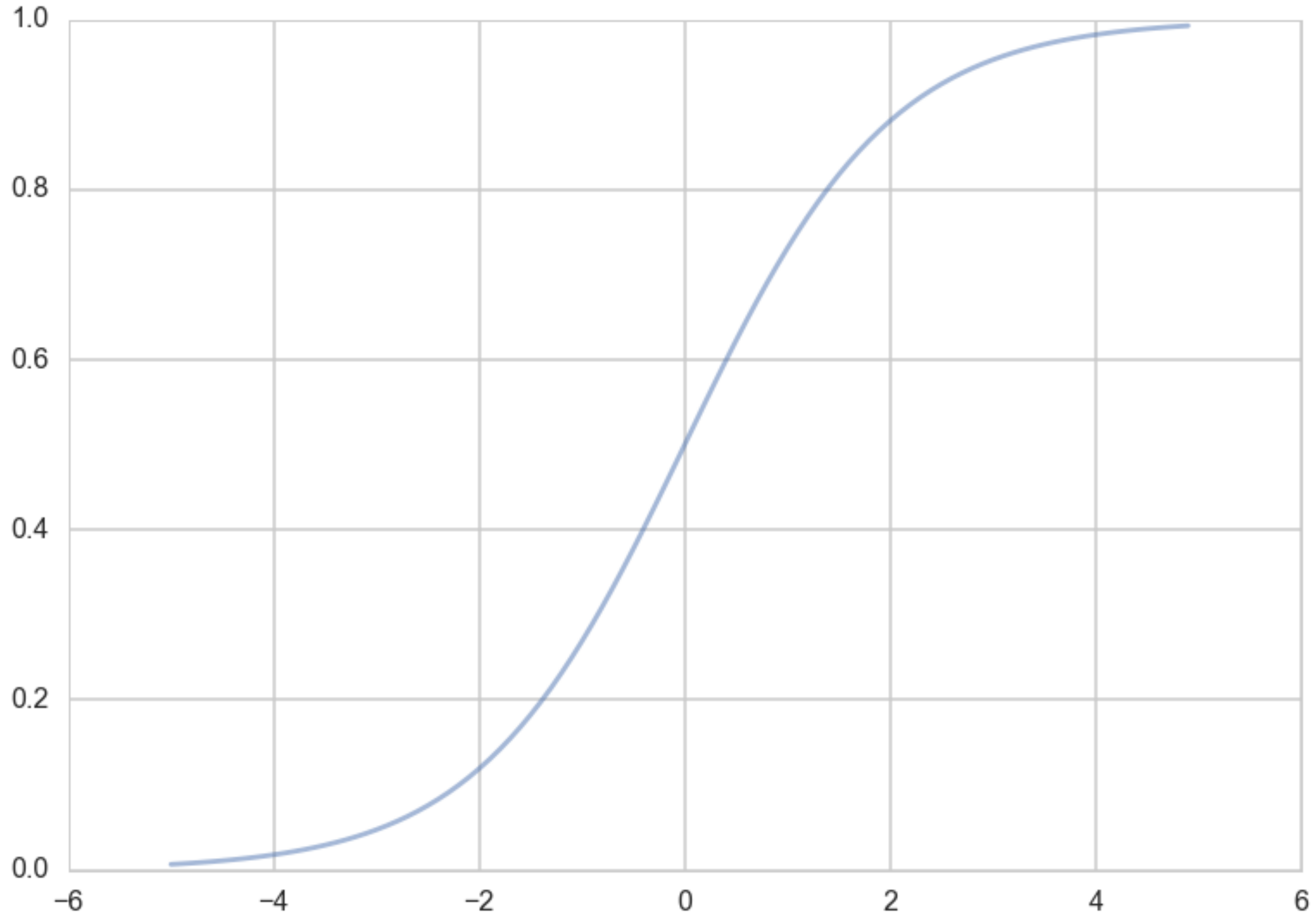
- example of a Generalized Linear Model (GLM)
- "Squeeze" linear regression through a **Sigmoid** function
- this bounds the output to be a probability
- What is the sampling Distribution?

Sigmoid function

This function is plotted below:

```
h = lambda z: 1./(1+np.exp(-z))
zs=np.arange(-5,5,0.1)
plt.plot(zs, h(zs), alpha=0.5);
```

Identify: $z = \mathbf{w} \cdot \mathbf{x}$. and $h(\mathbf{w} \cdot \mathbf{x})$ with the probability that the sample is a '1' ($y = 1$).



Then, the conditional probabilities of $y = 1$ or $y = 0$ given a particular sample's features $\mathbf{x}$ are:

$$P(y = 1|\mathbf{x}) = h(\mathbf{w} \cdot \mathbf{x})$$
$$P(y = 0|\mathbf{x}) = 1 - h(\mathbf{w} \cdot \mathbf{x}).$$

These two can be written together as

$$P(y|\mathbf{x}, \mathbf{w}) = h(\mathbf{w} \cdot \mathbf{x})^y (1 - h(\mathbf{w} \cdot \mathbf{x}))^{(1-y)}$$

BERNOULLI!!

Multiplying over the samples we get:

$$P(y|\mathbf{x}, \mathbf{w}) = P(\{y_i\}|\{\mathbf{x}_i\}, \mathbf{w}) = \prod_{y_i \in \mathcal{D}} P(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{y_i \in \mathcal{D}} h(\mathbf{w} \cdot \mathbf{x}_i)^{y_i} (1 - h(\mathbf{w} \cdot \mathbf{x}_i))^{(1-y_i)}$$

A noisy y is to imagine that our data $\mathcal{D}$ was generated from a joint probability distribution $P(x, y)$. Thus we need to model y at a given x, written as $P(y \mid x)$, and since $P(x)$ is also a probability distribution, we have:

$$P(x, y) = P(y \mid x)P(x),$$

Indeed its important to realize that a particular sample can be thought of as a draw from some "true" probability distribution.

maximum likelihood estimation maximises the **likelihood of the sample y**,

$$\mathcal{L} = P(y \mid \mathbf{x}, \mathbf{w}).$$

Again, we can equivalently maximize

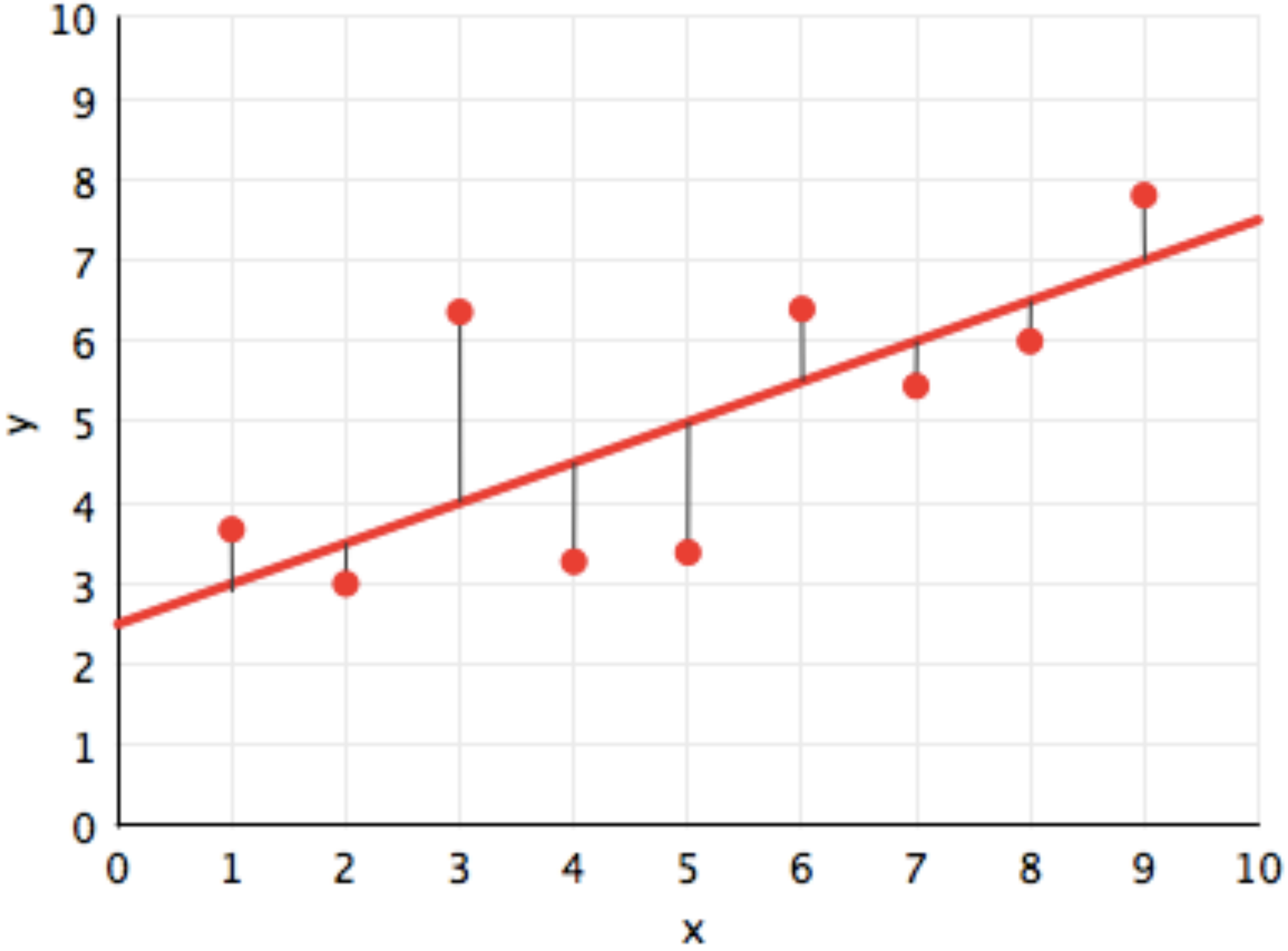
$$\ell = log(P(y \mid \mathbf{x}, \mathbf{w}))$$

Thus

$$
\begin{aligned}
\ell &= log\left(\prod_{y_i \in \mathcal{D}} h(\mathbf{w}\cdot\mathbf{x}_i)^{y_i}(1-h(\mathbf{w}\cdot\mathbf{x}_i))^{(1-y_i)}\right) \\
&= \sum_{y_i \in \mathcal{D}} log\left(h(\mathbf{w}\cdot\mathbf{x}_i)^{y_i}(1-h(\mathbf{w}\cdot\mathbf{x}_i))^{(1-y_i)}\right) \\
&= \sum_{y_i \in \mathcal{D}} log\, h(\mathbf{w}\cdot\mathbf{x}_i)^{y_i} + log\,(1-h(\mathbf{w}\cdot\mathbf{x}_i))^{(1-y_i)} \\
&= \sum_{y_i \in \mathcal{D}} (y_i log(h(\mathbf{w}\cdot\mathbf{x})) + (1-y_i)log(1-h(\mathbf{w}\cdot\mathbf{x})))
\end{aligned}
$$

REGRESSION

- how many dollars will you spend?
- what is your creditworthiness
- how many people will vote for Bernie t days before election
- use to predict probabilities for classification
- causal modeling in econometrics



From Bayesian Reasoning and Machine Learning, David Barber:

"A father decides to teach his young son what a sports car is. Finding it difficult to explain in words, he decides to give some examples. They stand on a motorway bridge and ... the father cries out 'that's a sports car!' when a sports car passes by. After ten minutes, the father asks his son if he's understood what a sports car is. The son says, 'sure, it's easy'. An old red VW Beetle passes by, and the son shouts – 'that's a sports car!'. Dejected, the father asks – 'why do you say that?'. 'Because all sports cars are red!', replies the son."

HYPOTHESIS SPACES

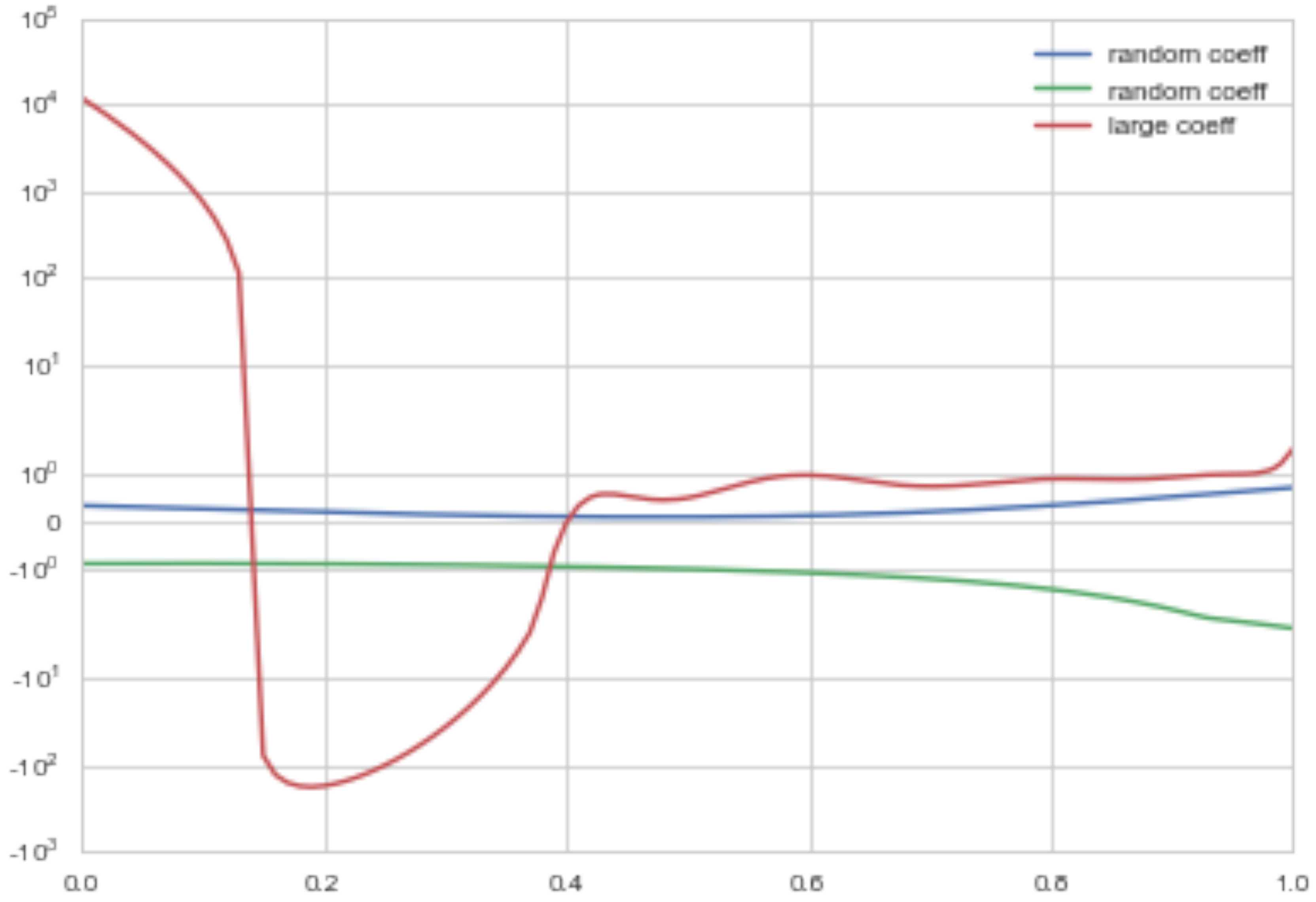
A polynomial looks so:

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

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

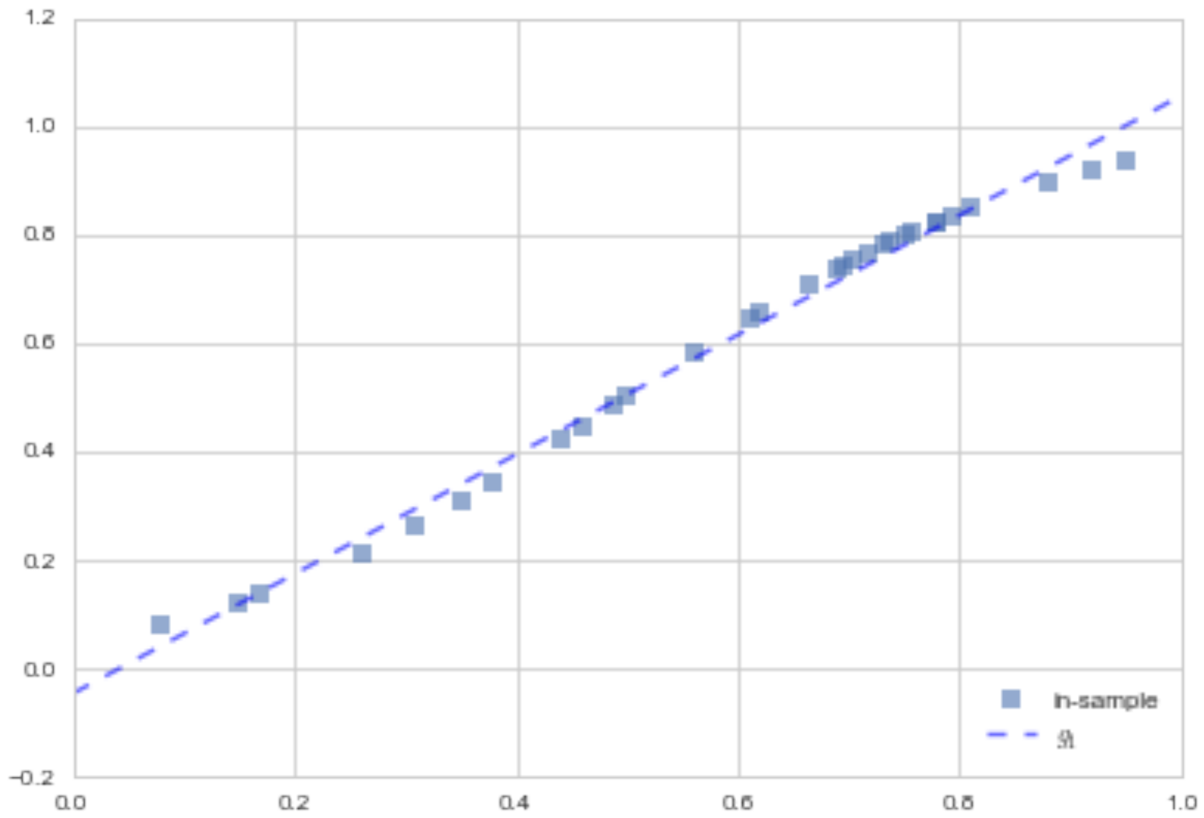
$$\mathcal{H}_1 : h_1(x) = \theta_0 + \theta_1 x$$

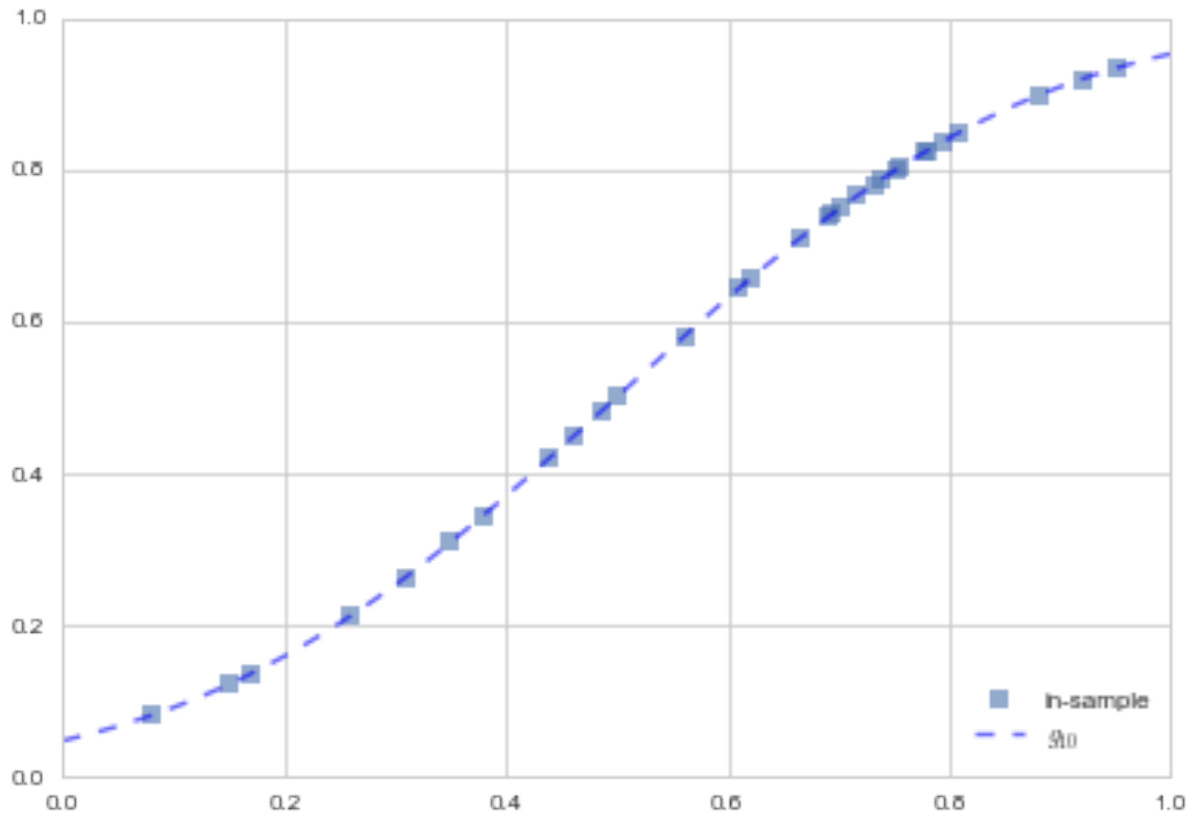
$$\mathcal{H}_{20} : h_{20}(x) = \sum_{i=0}^{20} \theta_i x^i$$



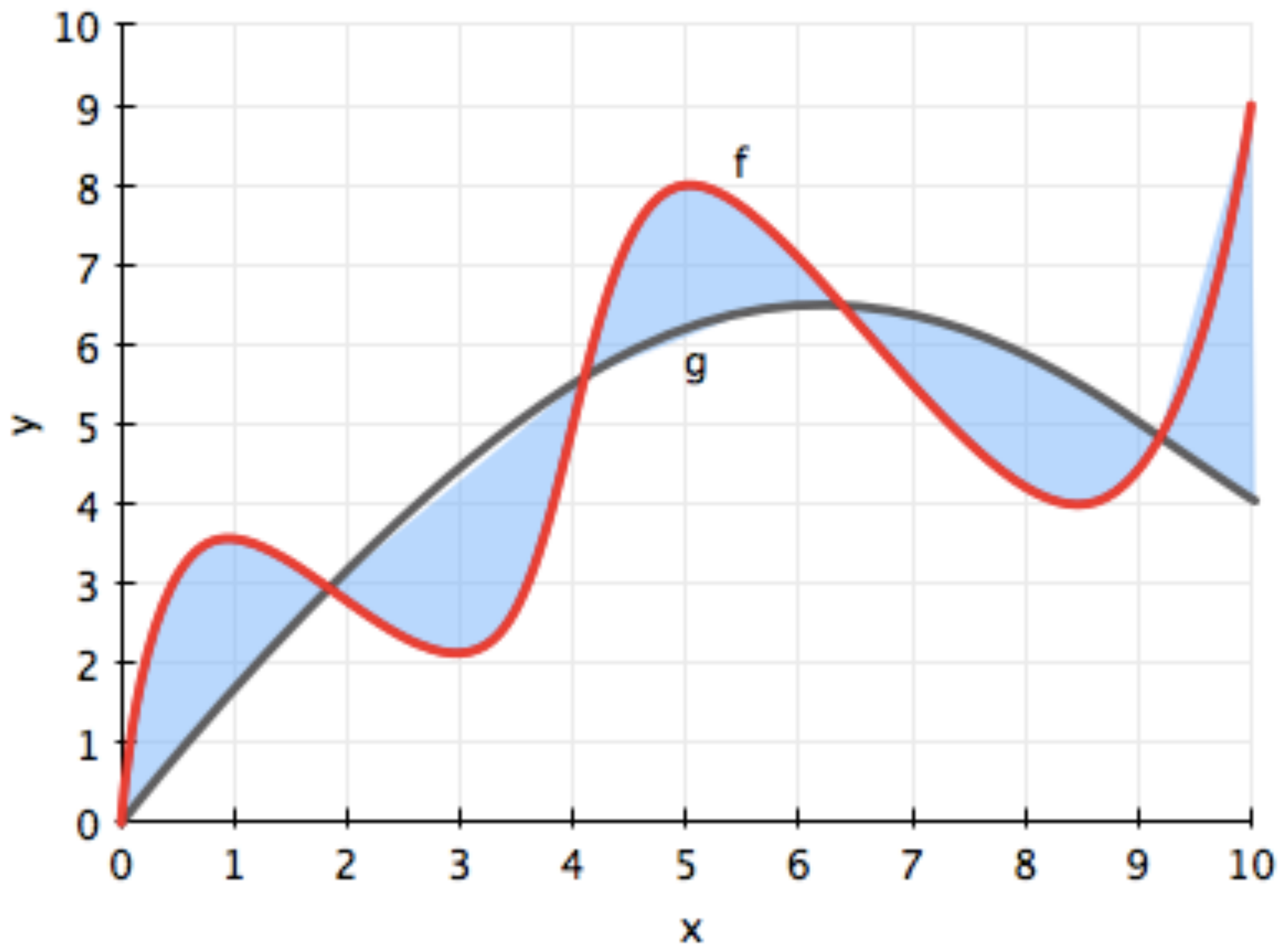
Approximation: Learning without noise

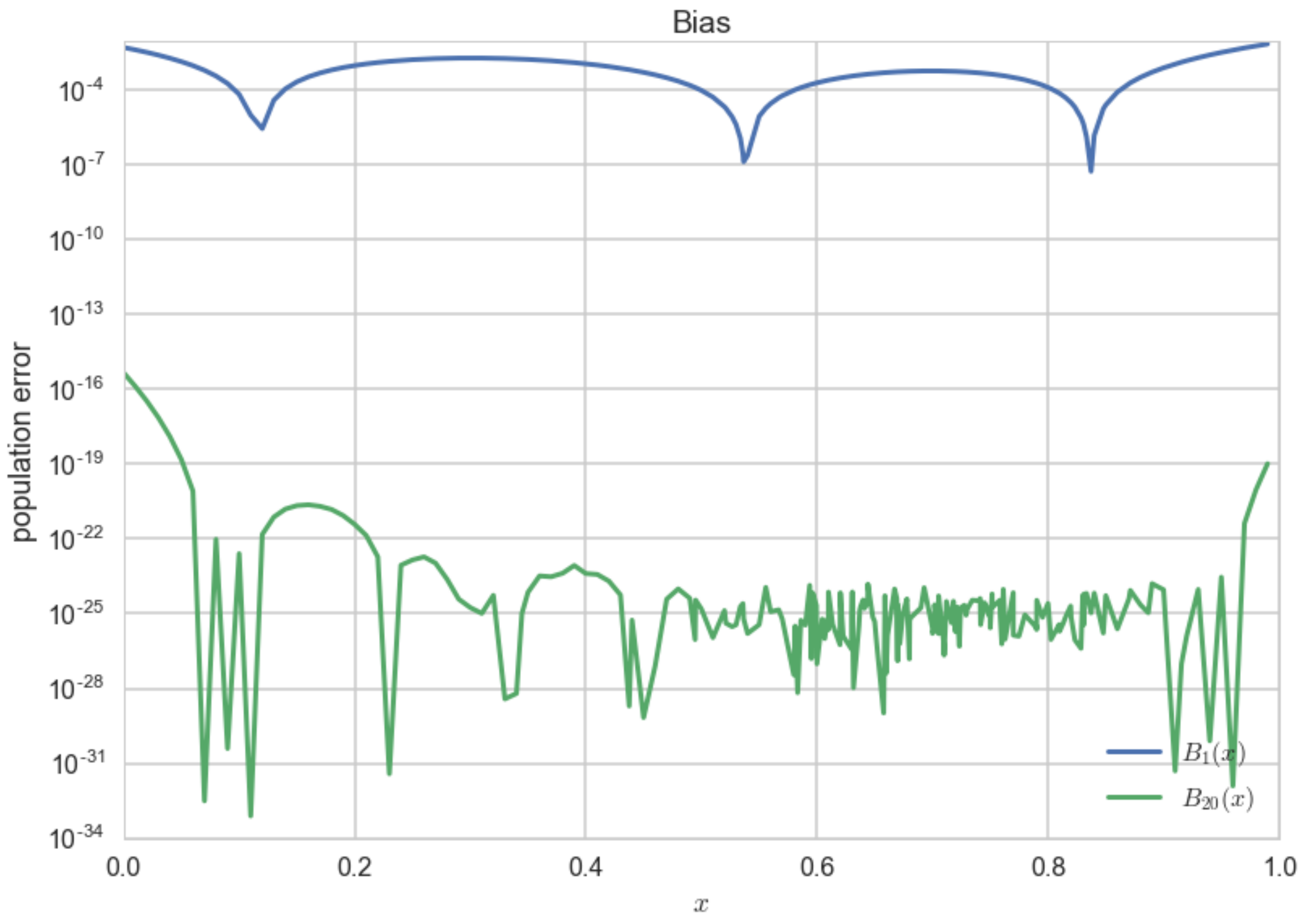
30 points of data. Which fit is better? Line in $\mathcal{H}_1$ or curve in $\mathcal{H}_{20}$?



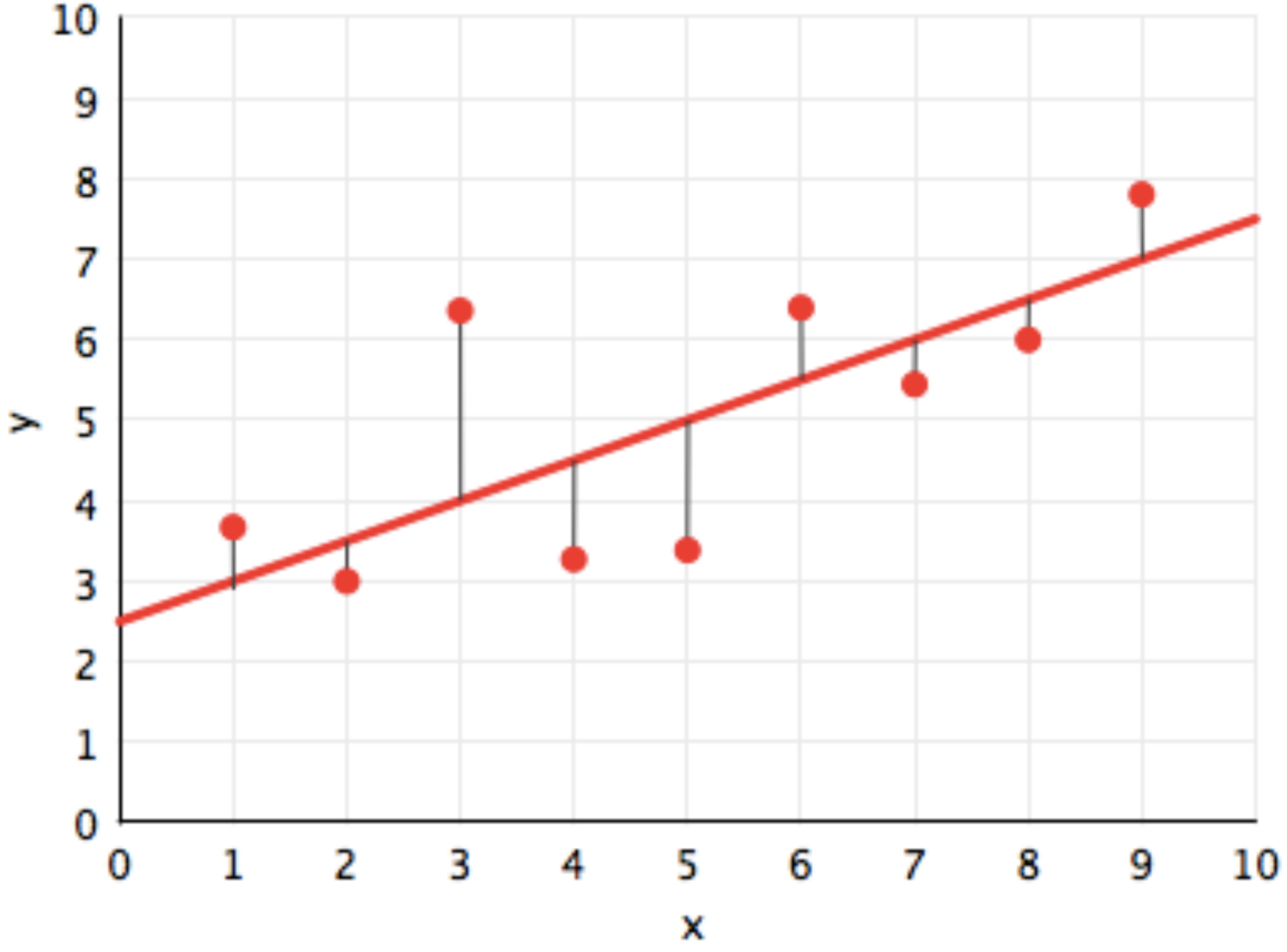


Bias or Mis-specification Error





RISK: What does it mean to FIT?



Minimize distance from the line?

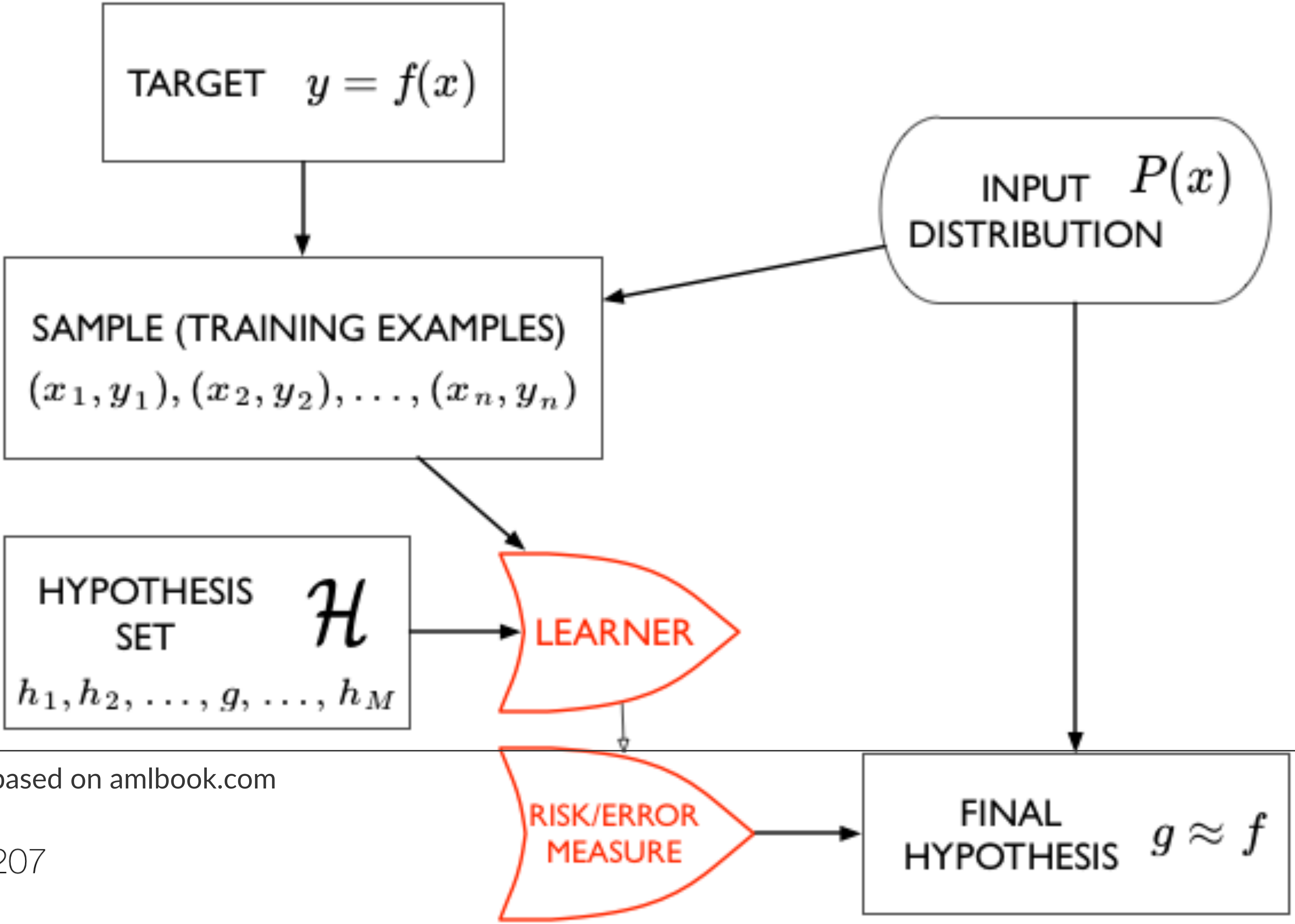
$$R_{\mathcal{D}}(h_1(x)) = \frac{1}{N} \sum_{y_i \in \mathcal{D}} (y_i - h_1(x_i))^2$$

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.

*



* image based on amlbook.com

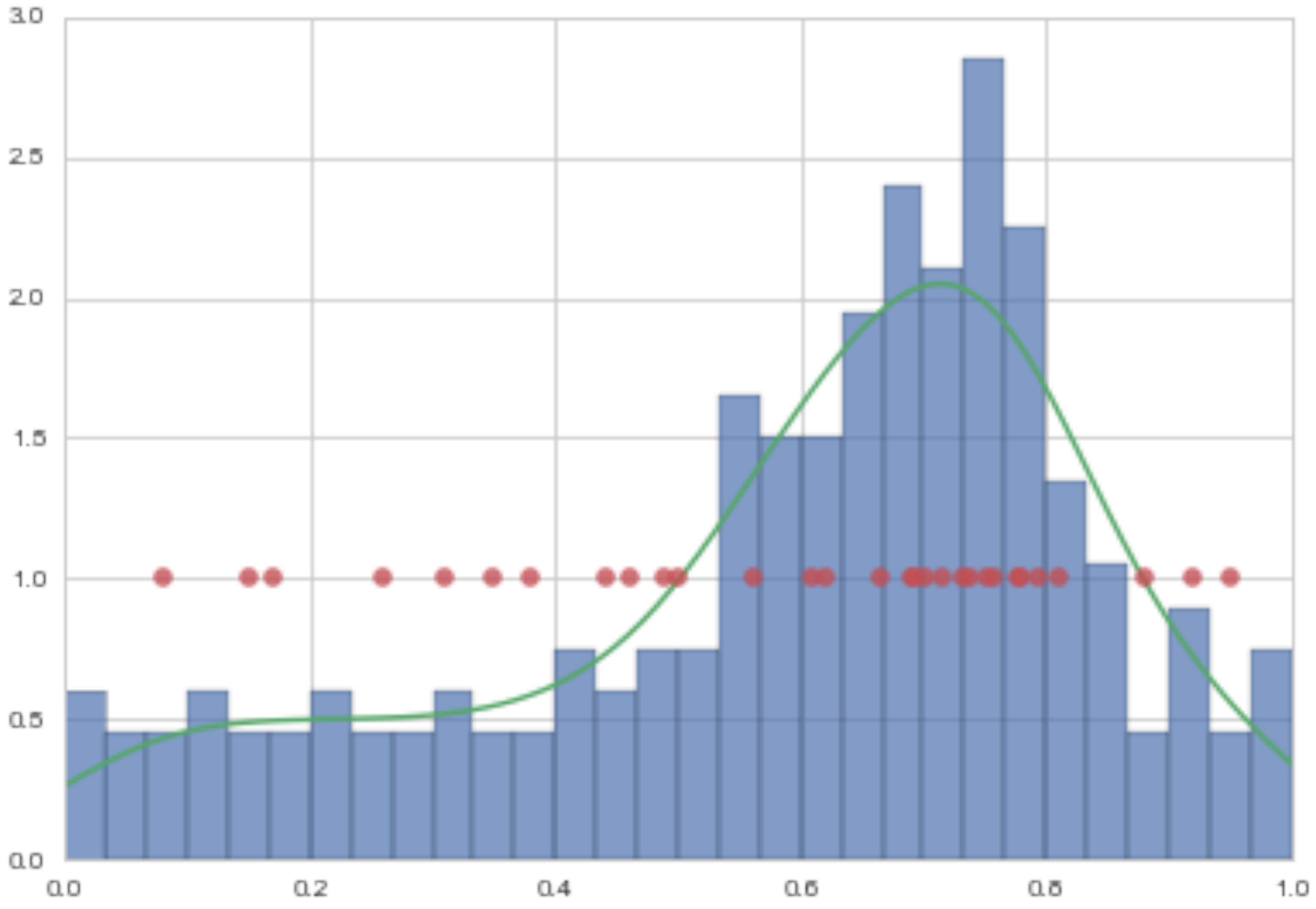
SAMPLE vs POPULATION

Want: $R_{out}(h) = E_{p(x)}[(h(x) - f(x))^2] = \int dx p(x) (h(x) - f(x))^2$

LLN:

$$R_{out}(h) = \lim_{n \to \infty} \frac{1}{n} \sum_{x_i \sim p(x)} (h(x_i) - f(x_i))^2 = \lim_{n \to \infty} \frac{1}{n} \sum_{x_i \sim p(x)} (h(x_i) - y_i)^2$$

$\mathcal{D}$ representative ($\mathcal{D} \sim p(x)$) $\implies \mathcal{R}_{\mathcal{D}}(h) = \sum_{x_i \in \mathcal{D}} (h(x_i) - y_i)^2$



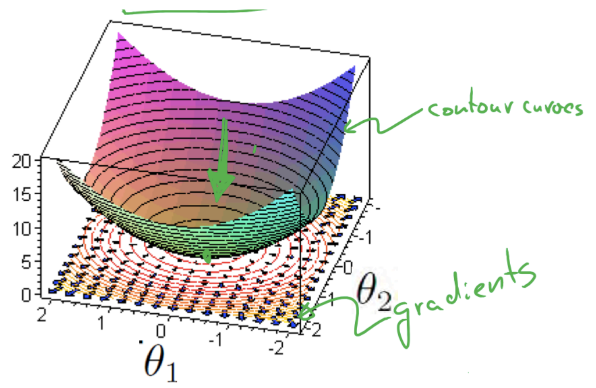
Statement of the Learning Problem

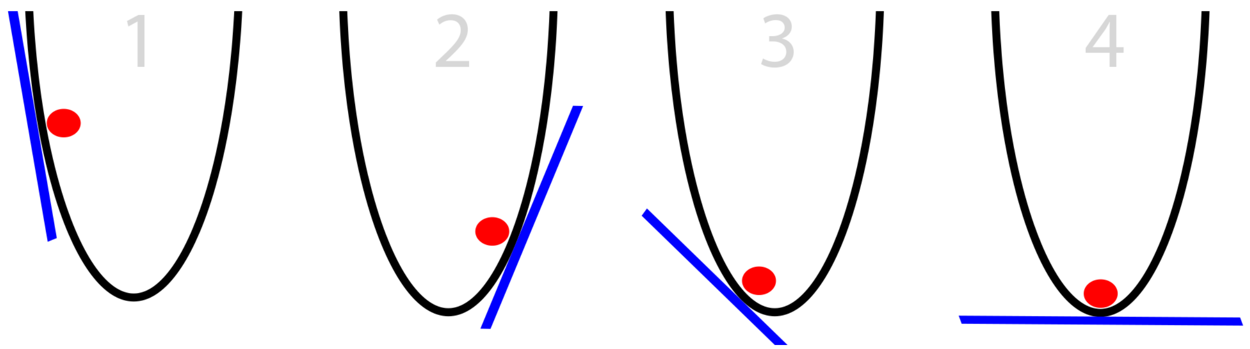
The sample must be representative of the population!

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

A: Empirical risk estimates in-sample risk.
B: Thus the out of sample risk is also small.

CONVEX MINIMIZATION



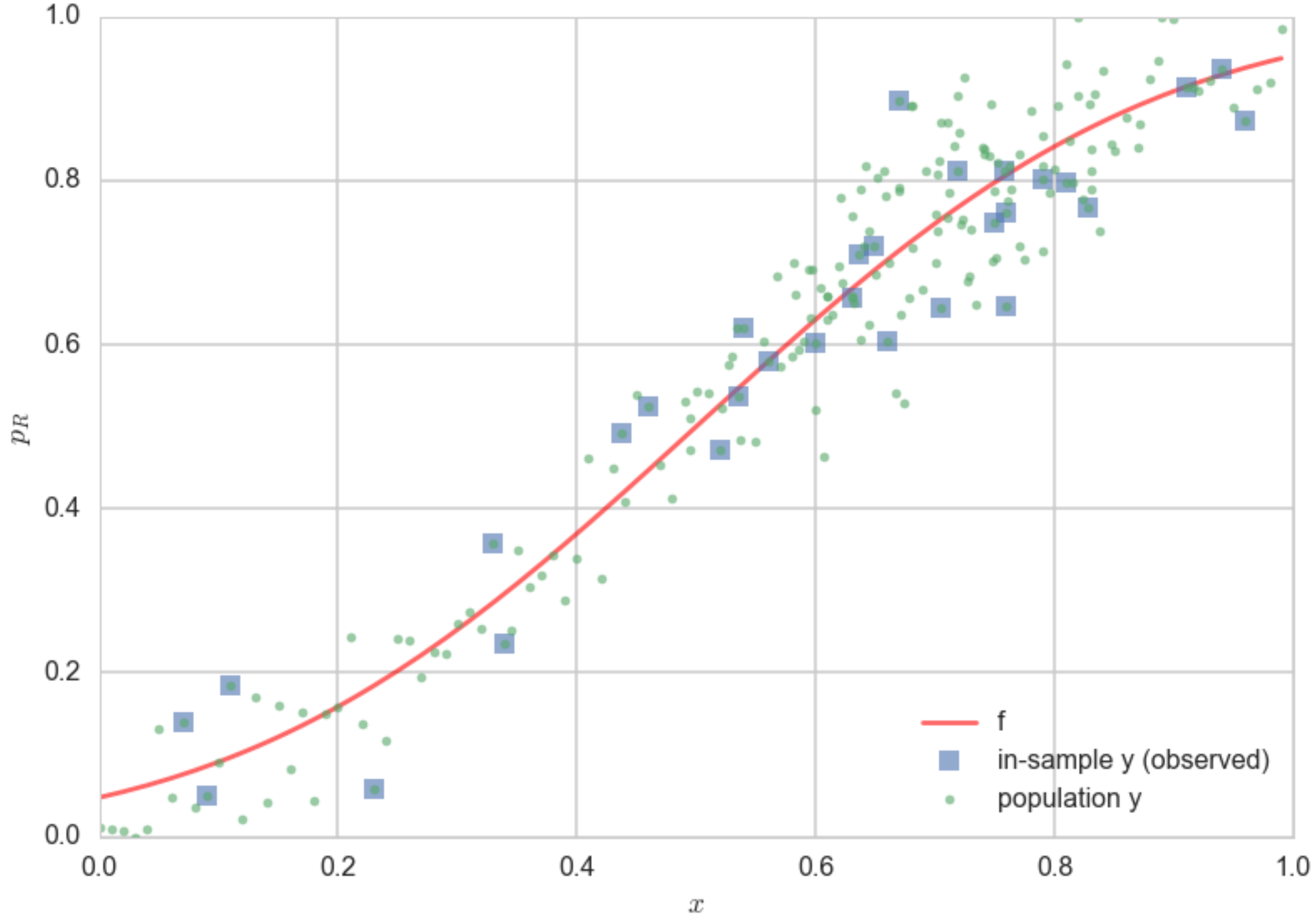


In general one can use gradient descent .

For linear-regression, one can however just do this using matrix algebra.

Image From Nando-deFreitas Deep Learning Course 2015

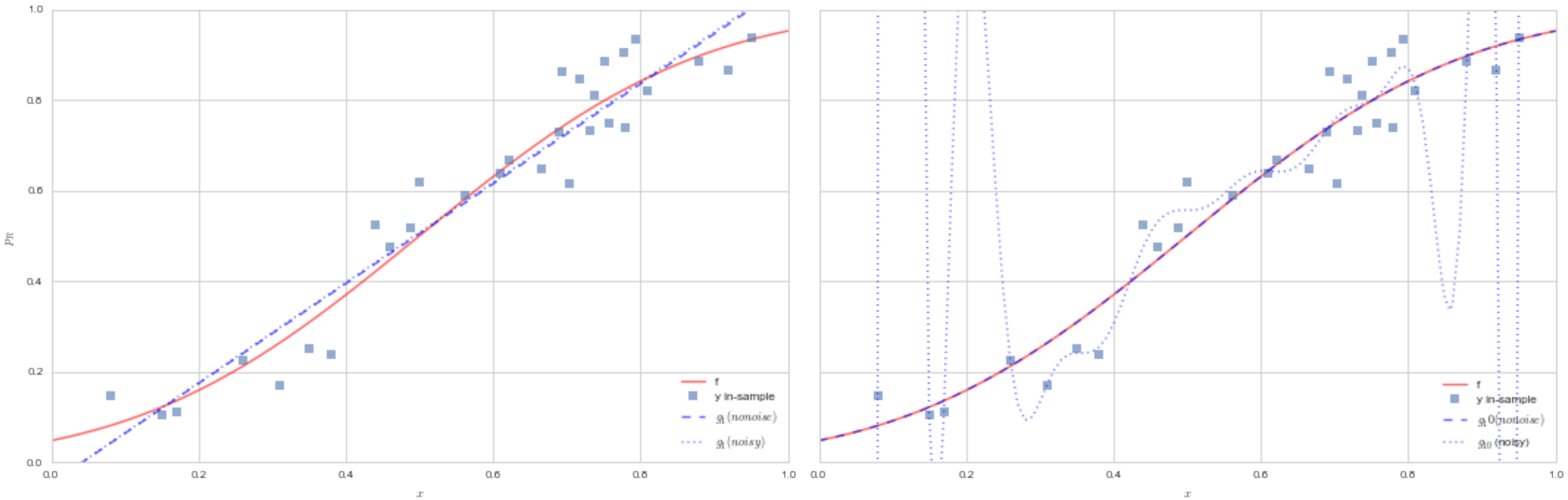
THE REAL WORLD HAS NOISE



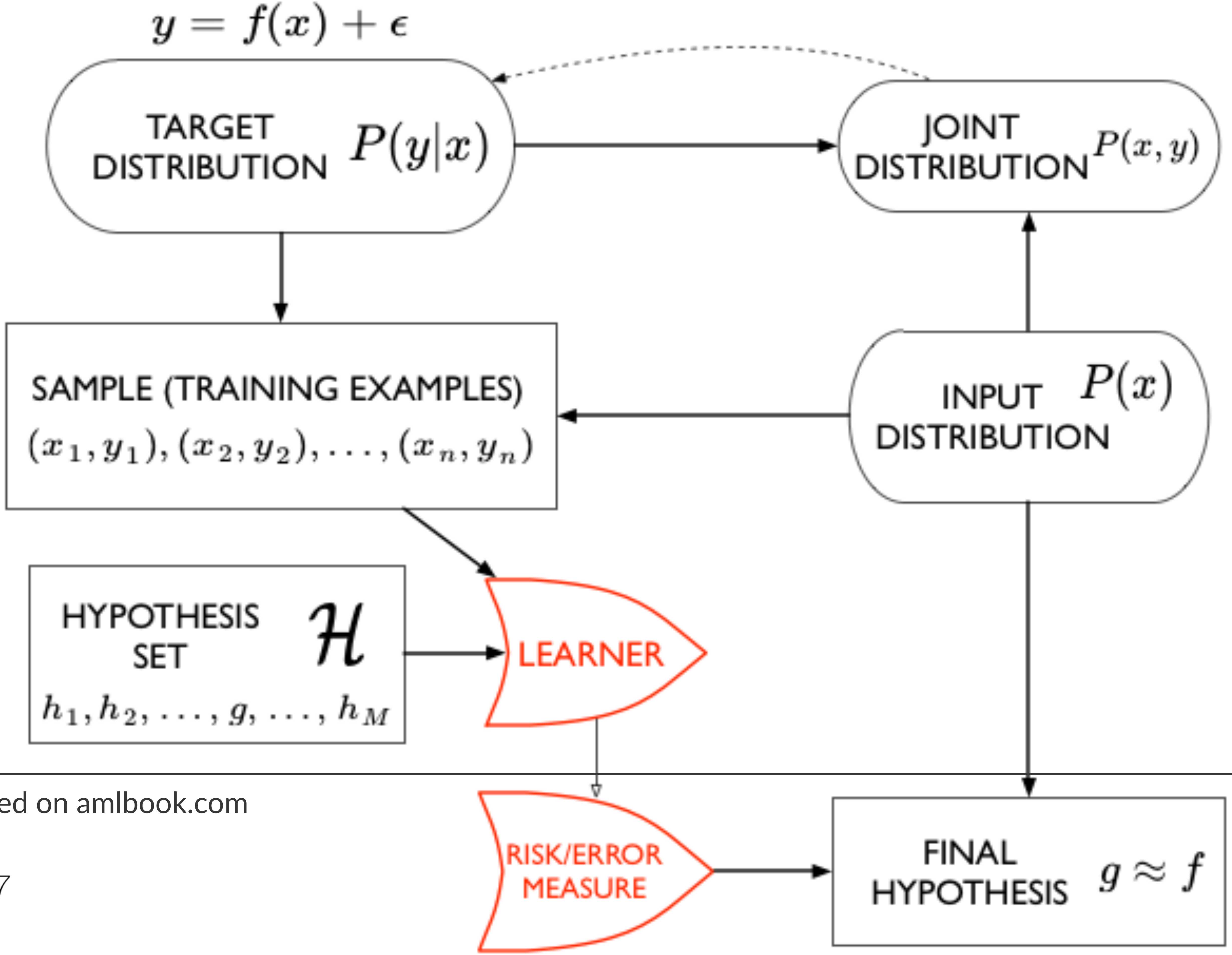
THE REAL WORLD HAS NOISE

Which fit is better now?

The line or the curve?



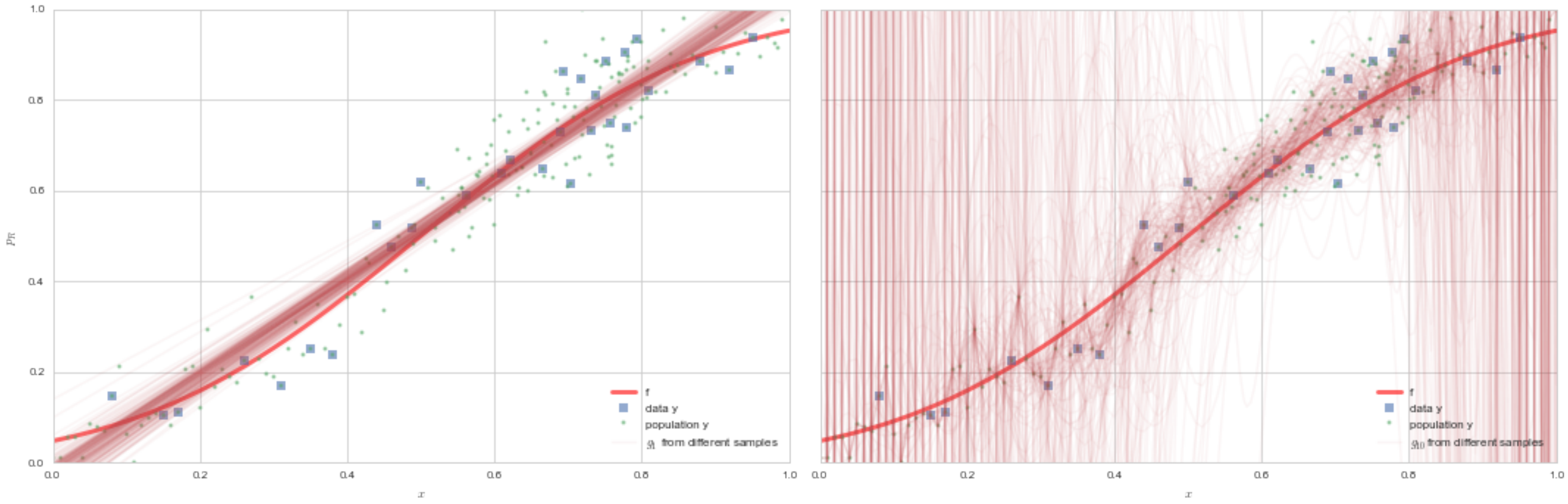
*



* image based on amlbook.com

UNDERFITTING (Bias)

vs OVERFITTING (Variance)



Every model has Bias and Variance

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

Fit hypothesis $h = g_{\mathcal{D}}$, where $\mathcal{D}$ is our training sample.

Define:

$$\langle R \rangle = \int dy dx \, p(x, y)(h(x) - y)^2 = \int dy dx p(y \mid x) p(x)(h(x) - y)^2.$$

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

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

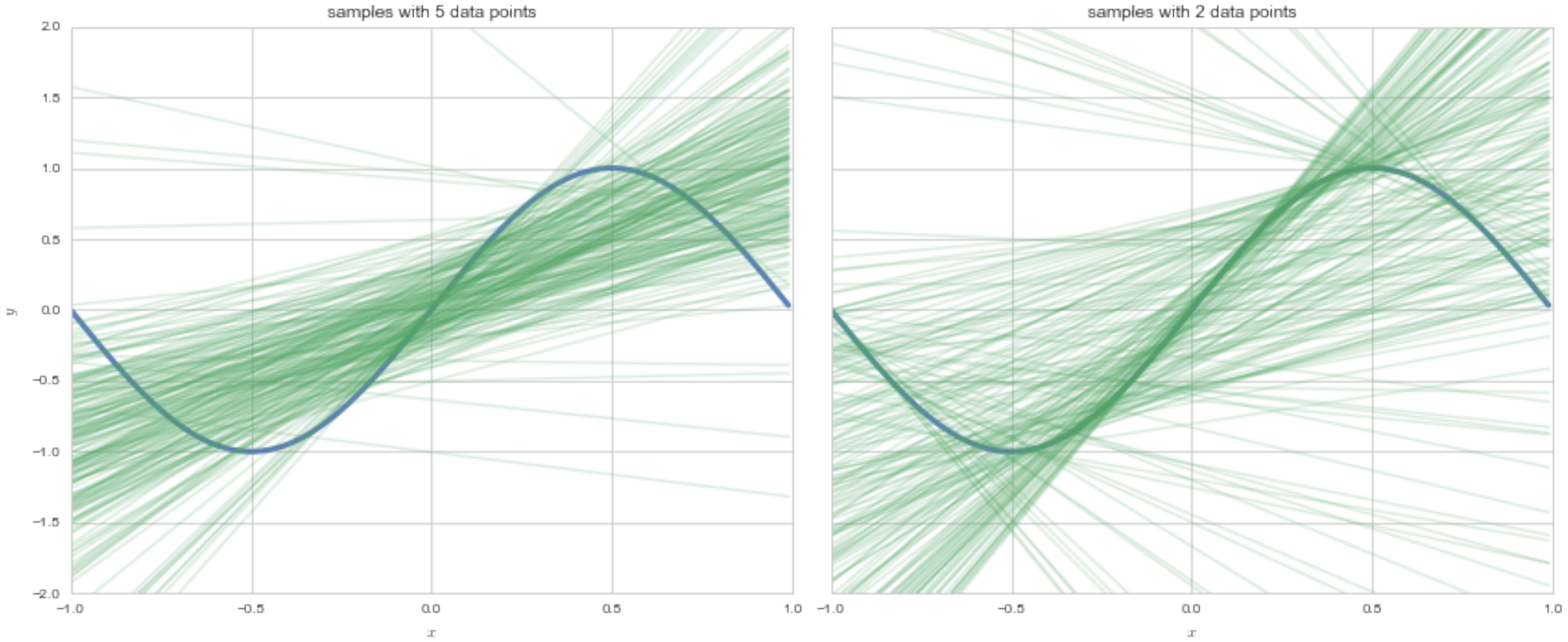
Then,

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

This is the bias variance decomposition for regression.

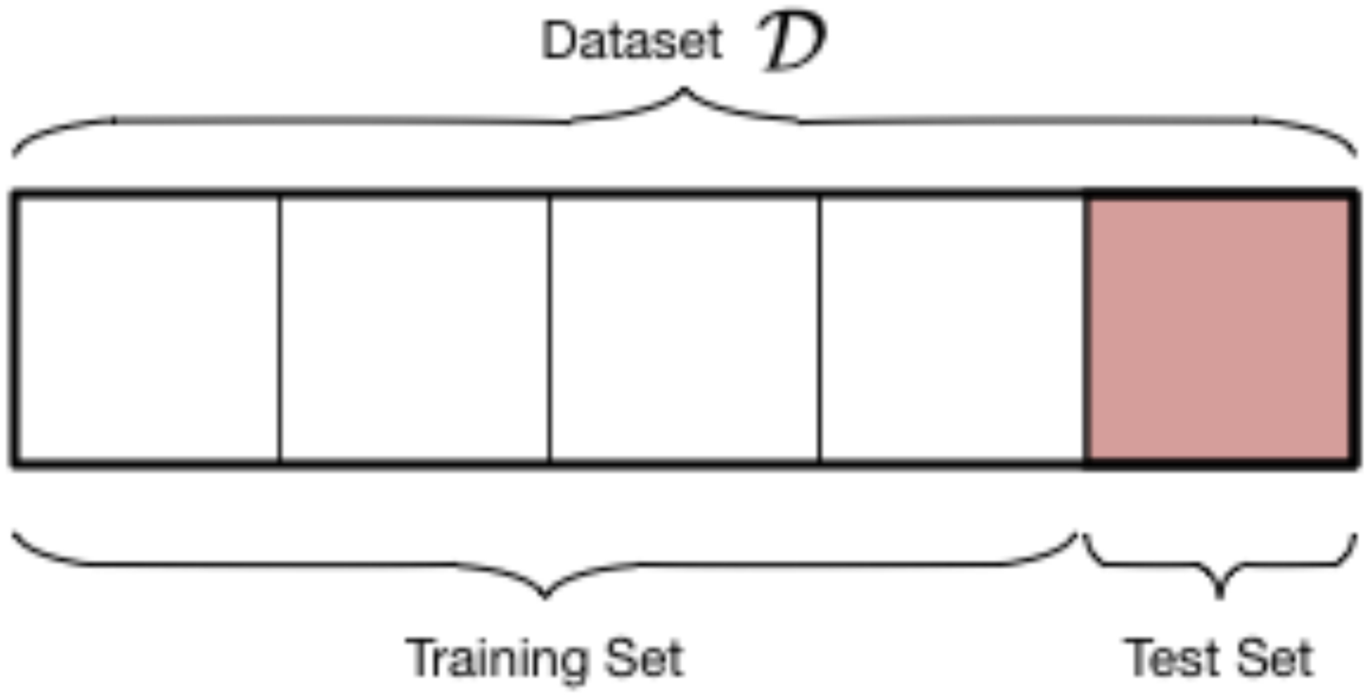
- first term is **variance**, squared error of the various fit g's from the average g, the hairiness.
- second term is **bias**, how far the average g is from the original f this data came from.
- third term is the **stochastic noise**, minimum error that this model will always have.

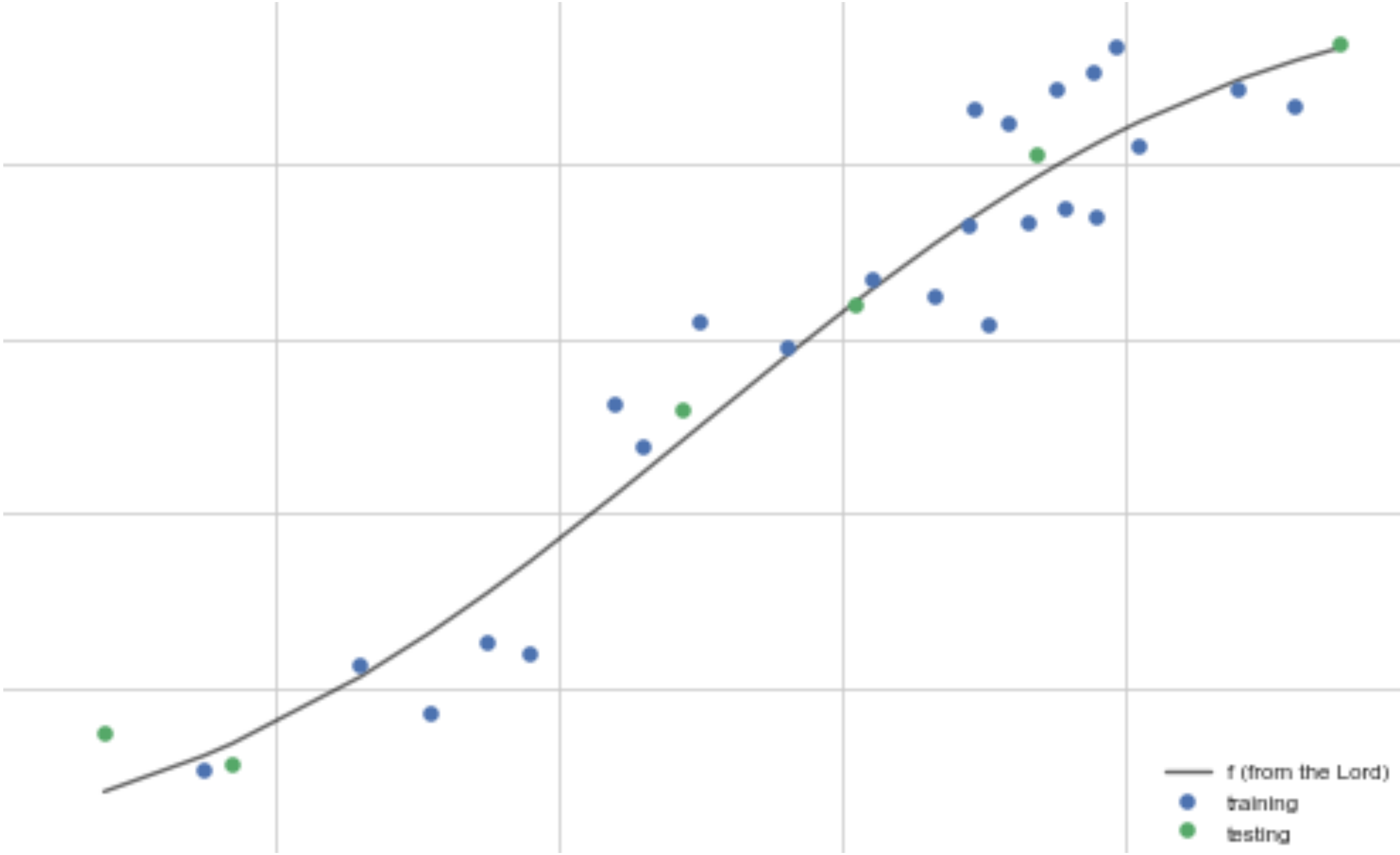
DATA SIZE MATTERS: straight line fits to a sine curve



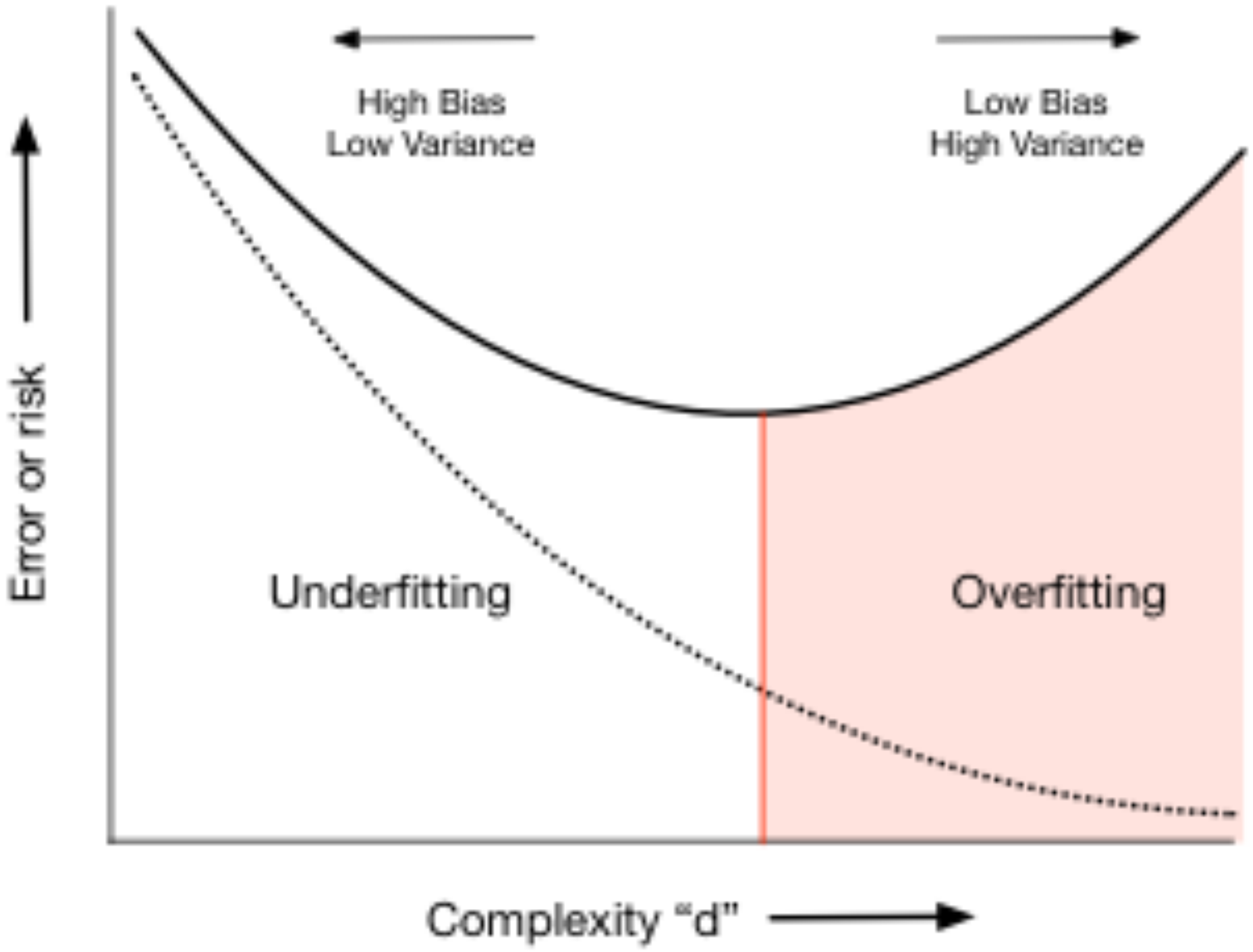
Corollary: Must fit simpler models to less data!

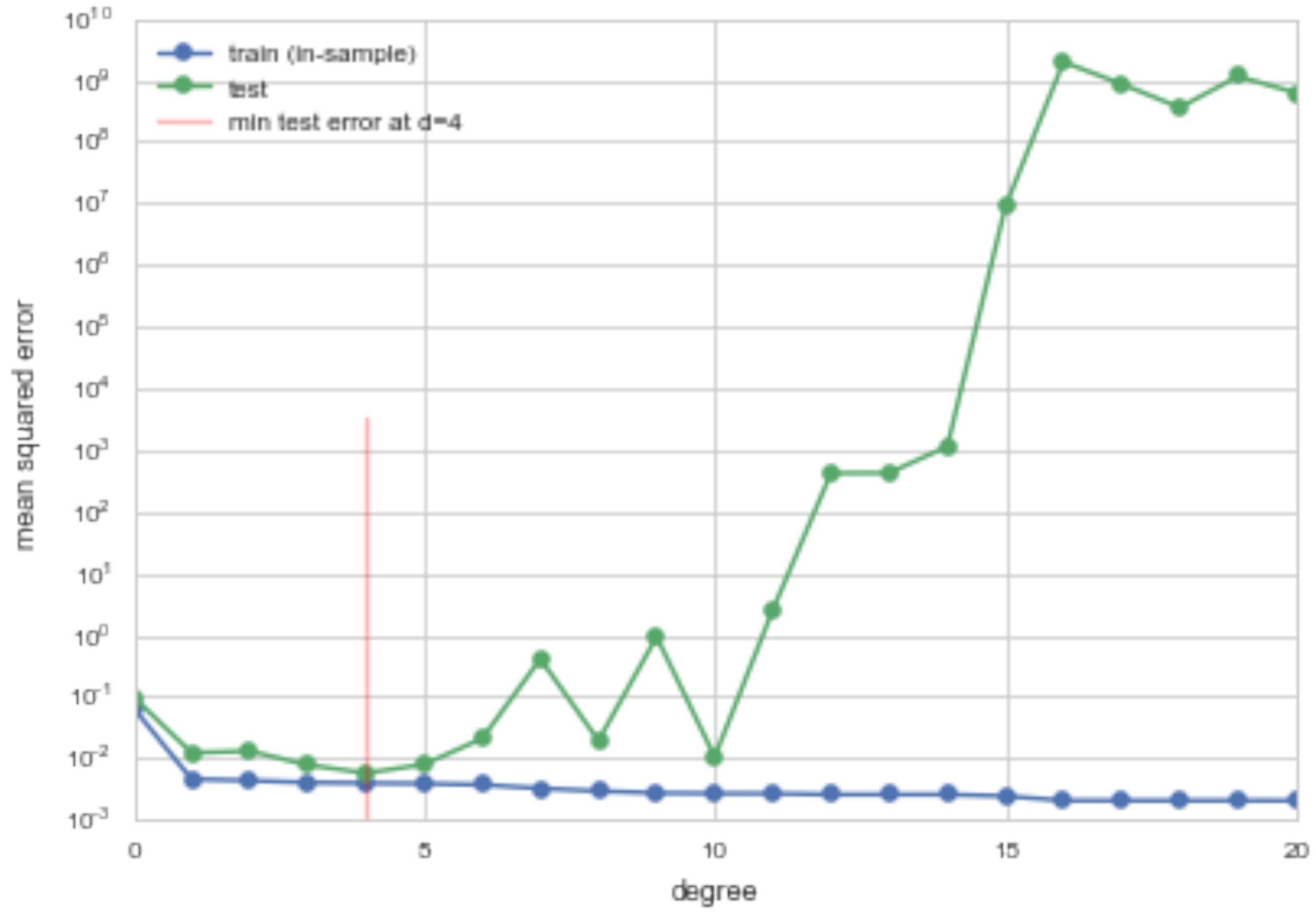
TRAIN AND TEST





BALANCE THE COMPLEXITY





Is this still 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**.

Hoeffding's inequality

population fraction μ, sample drawn with replacement, fraction ν:

$$P(|\nu - \mu| > \epsilon) \leq 2e^{-2\epsilon^2 N}$$

For hypothesis h, identify 1 with $h(x_i) \neq f(x_i)$ at sample x_i. Then μ, ν are population/sample error rates. Then,

$$P(|R_{in}(h) - R_{out}(h)| > \epsilon) \leq 2e^{-2\epsilon^2 N}$$

- Hoeffding inequality holds ONCE we have picked a hypothesis h, as we need it to label the 1 and 0s.
- But over the training set we one by one pick all the models in the hypothesis space
- best fit g is among the h in $\mathcal{H}$, g must be h_1 OR h_2 OR....Say **effectively** M such choices:

$$P(|R_{in}(g) - R_{out}(g)| \geq \epsilon) <= \sum_{h_i \in \mathcal{H}} P(|R_{in}(h_i) - R_{out}(h_i)| \geq \epsilon) <= 2\,M\,e^{-2\epsilon^2 N}$$

Hoeffding, repharased:

Now let $\delta = 2\,M\,e^{-2\epsilon^2 N}$.

Then, **with probability** $1-\delta$:

$$R_{out} <= R_{in} + \sqrt{\frac{1}{2N} ln(\frac{2M}{\delta})}$$

For finite effective hypothesis set size M, $R_{out} \sim R_{in}$ as N larger..

Training vs Test

- training error approximates out-of-sample error slowly
- is test set just another sample like the training sample?
- key observation: test set is looking at only one hypothesis because the fitting is already done on the training set. So $M = 1$ for this sample!

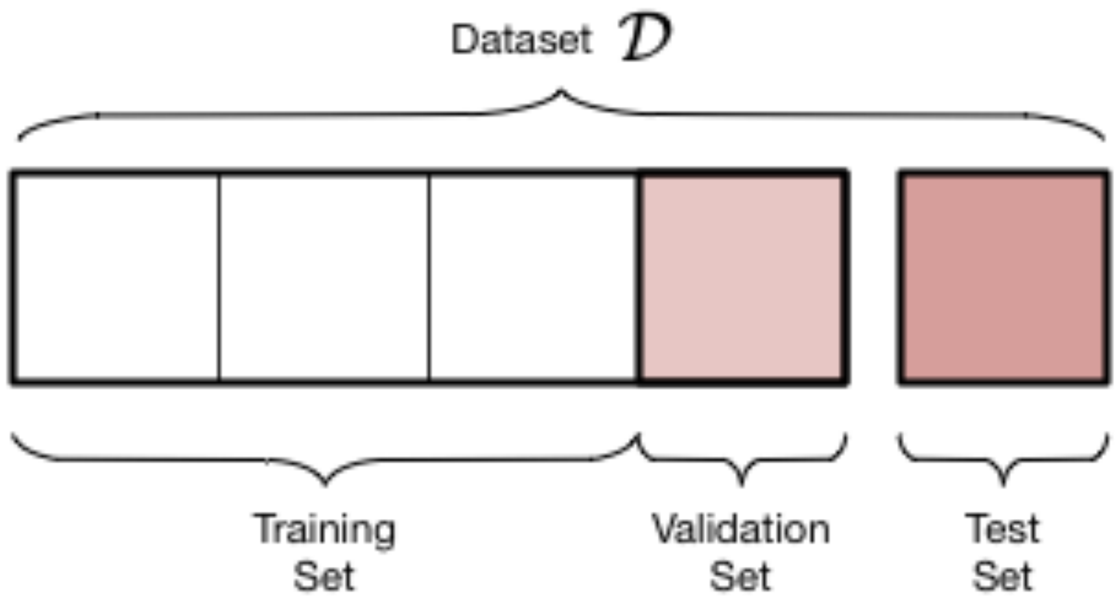
$$R_{out} <= R_{in} + \sqrt{\frac{1}{2N_{test}} ln(\frac{2}{\delta})}$$

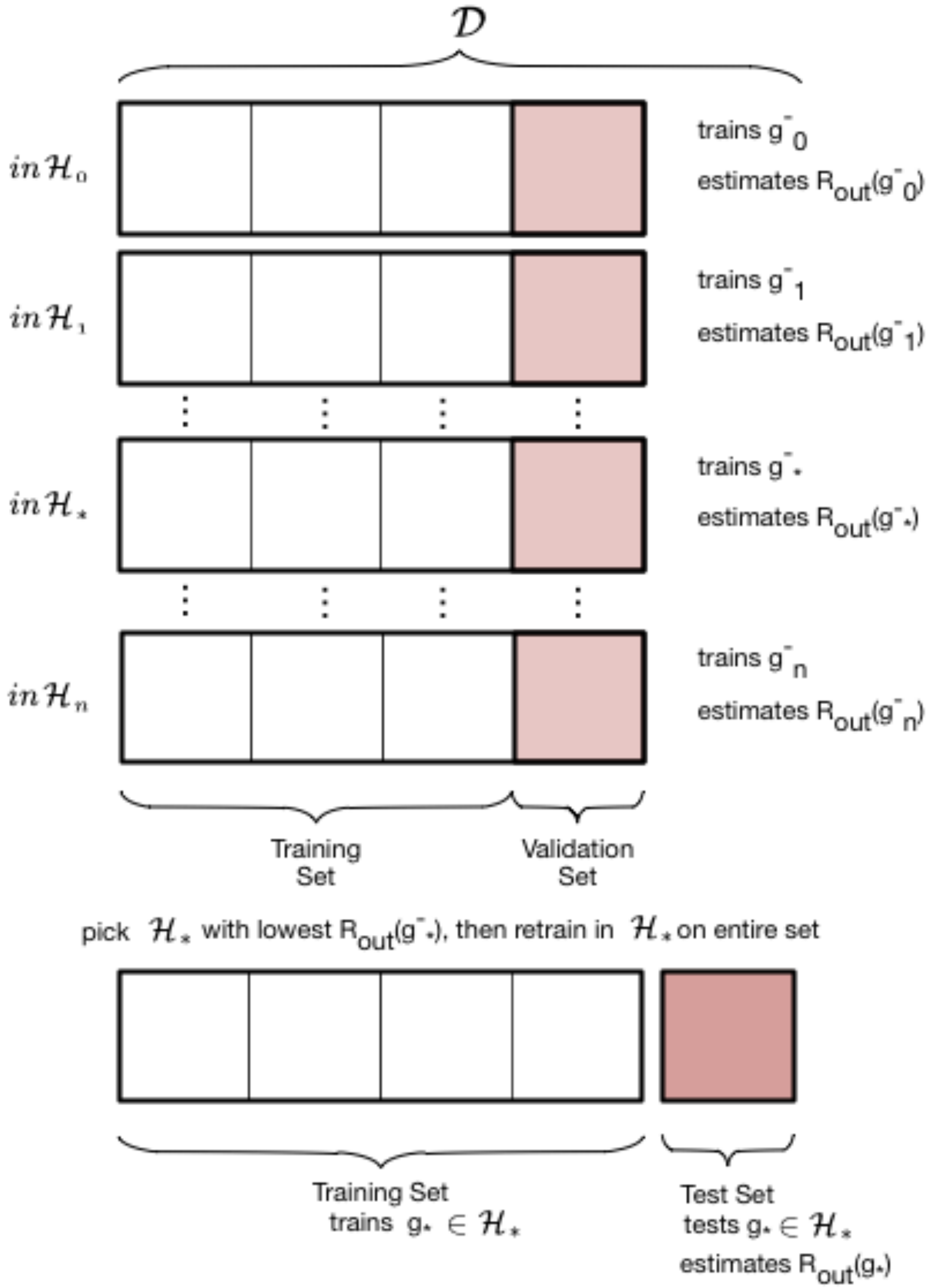
Training vs Test

- the test set does not have an optimistic bias like the training set(thats why the larger effective M factor)
- once you start fitting for things like d on the test set, you cant call it a test set any more since we lose tight guarantee.
- test set has a cost of less data in the training set and must thus fit a less complex model.

VALIDATION

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





If we dont fit a hyperparameter

- first assume that the validation set is acting like a test set.
- validation risk or error is an unbiased estimate of the out of sample risk.
- Hoeffding bound for a validation set is then identical to that of the test set.

usually we want to fit a hyperparameter

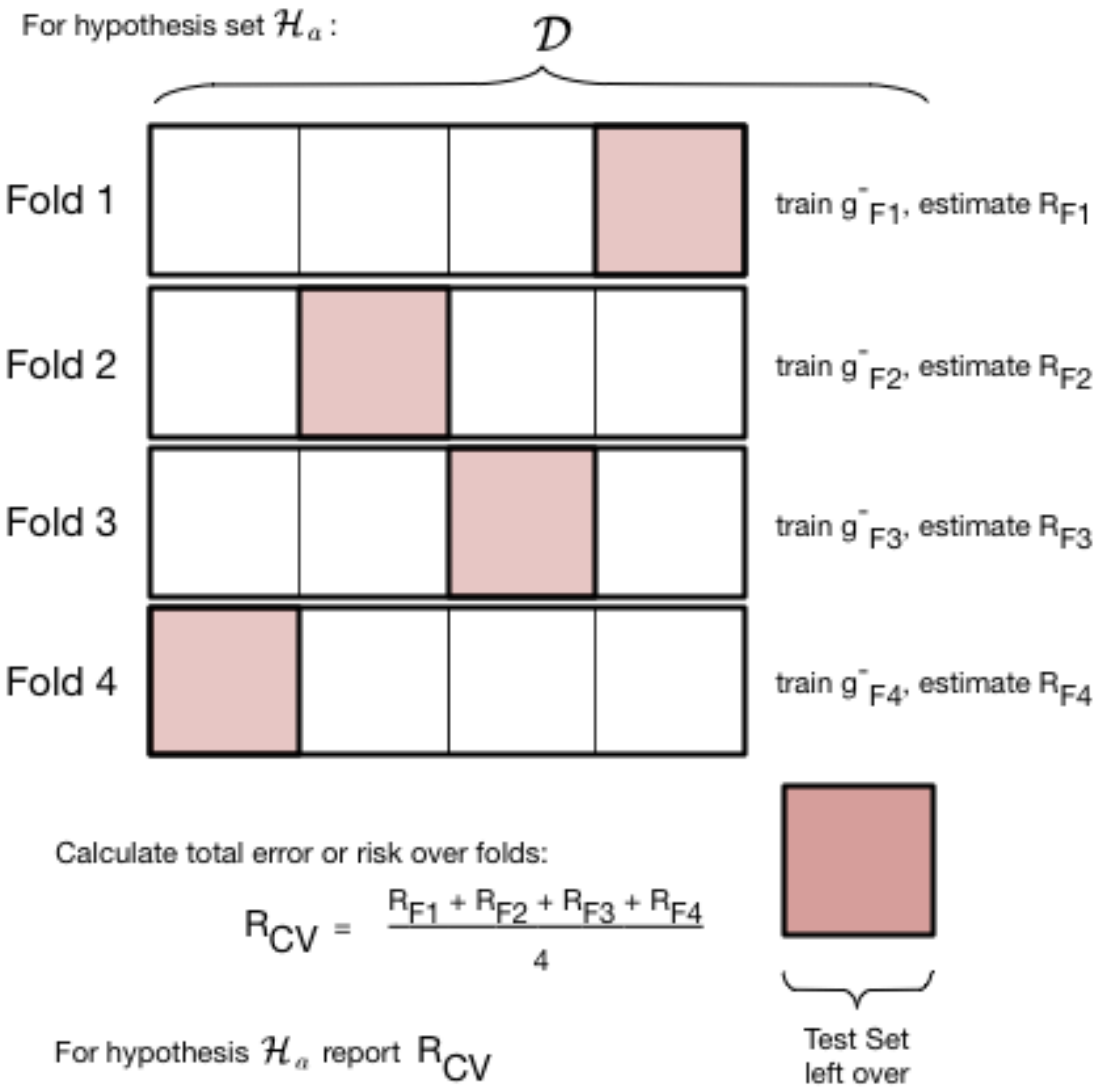
- we **wrongly** already attempted to do on our previous test set.
- choose the d, g^* combination with the lowest validation set risk.
- $R_{val}(g^{-*}, d^*)$ has an optimistic bias since d effectively fit on validation set
- its Hoeffding bound must now take into account the grid-size as the effective size of the hypothesis space.

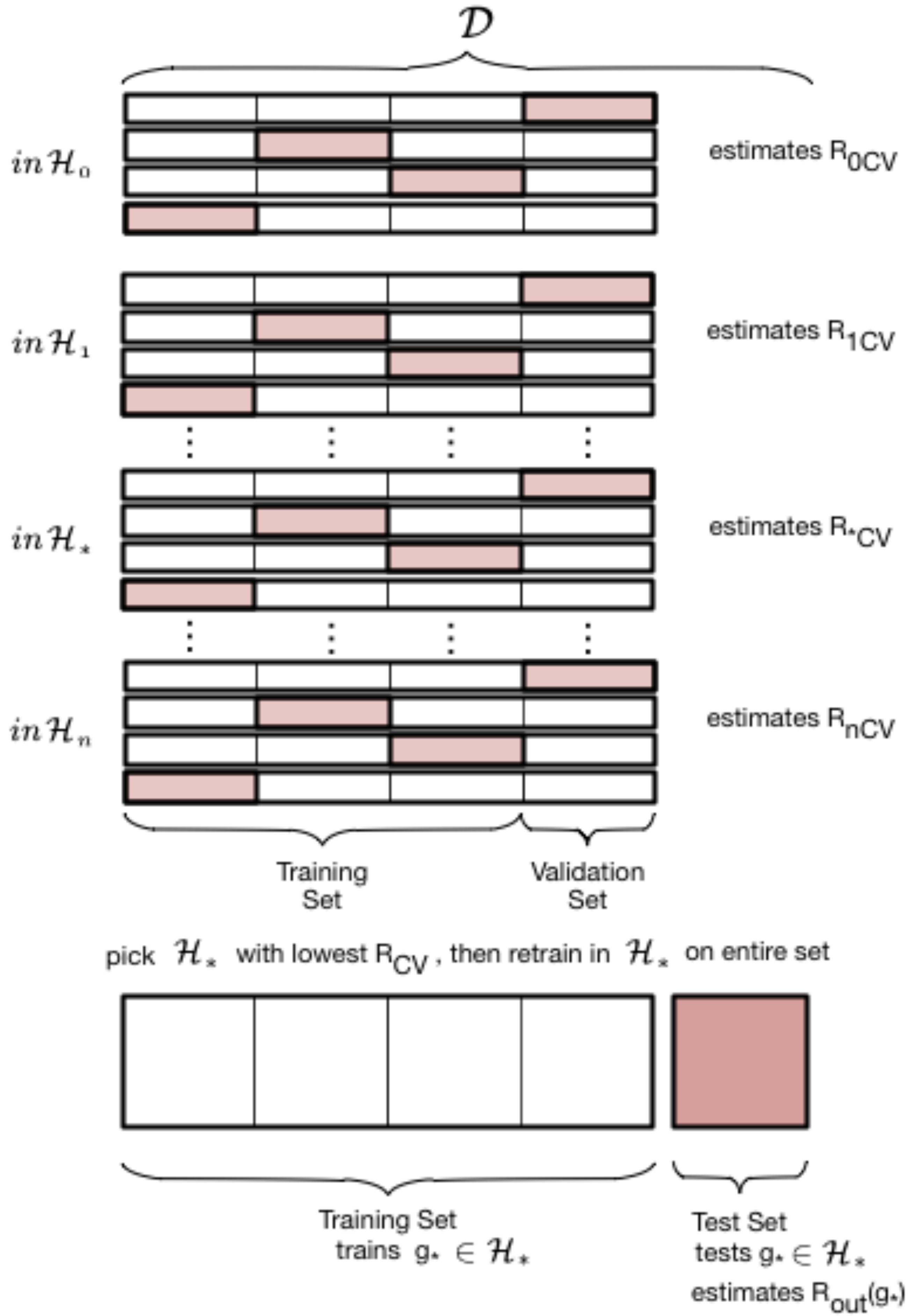
- this size from hyperparameters is typically a smaller size than that from parameters.

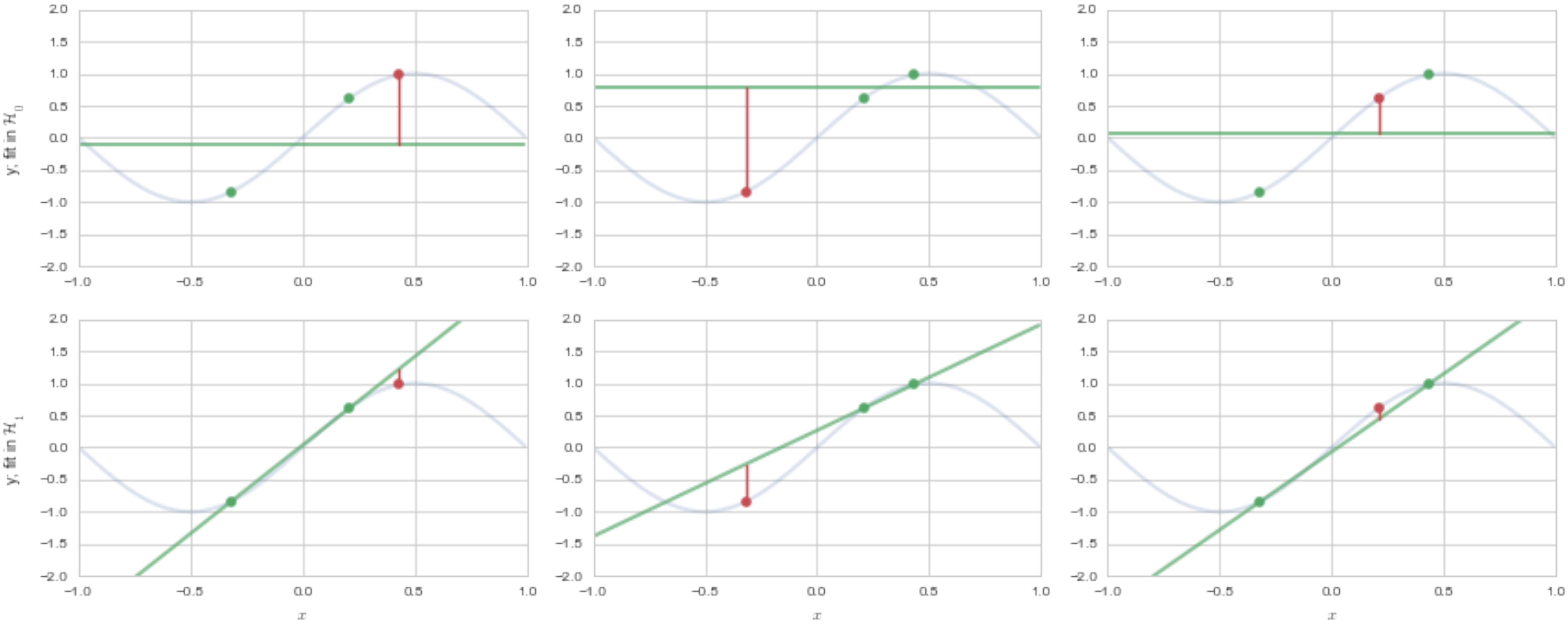
Retrain on entire set!

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

CROSS-VALIDATION





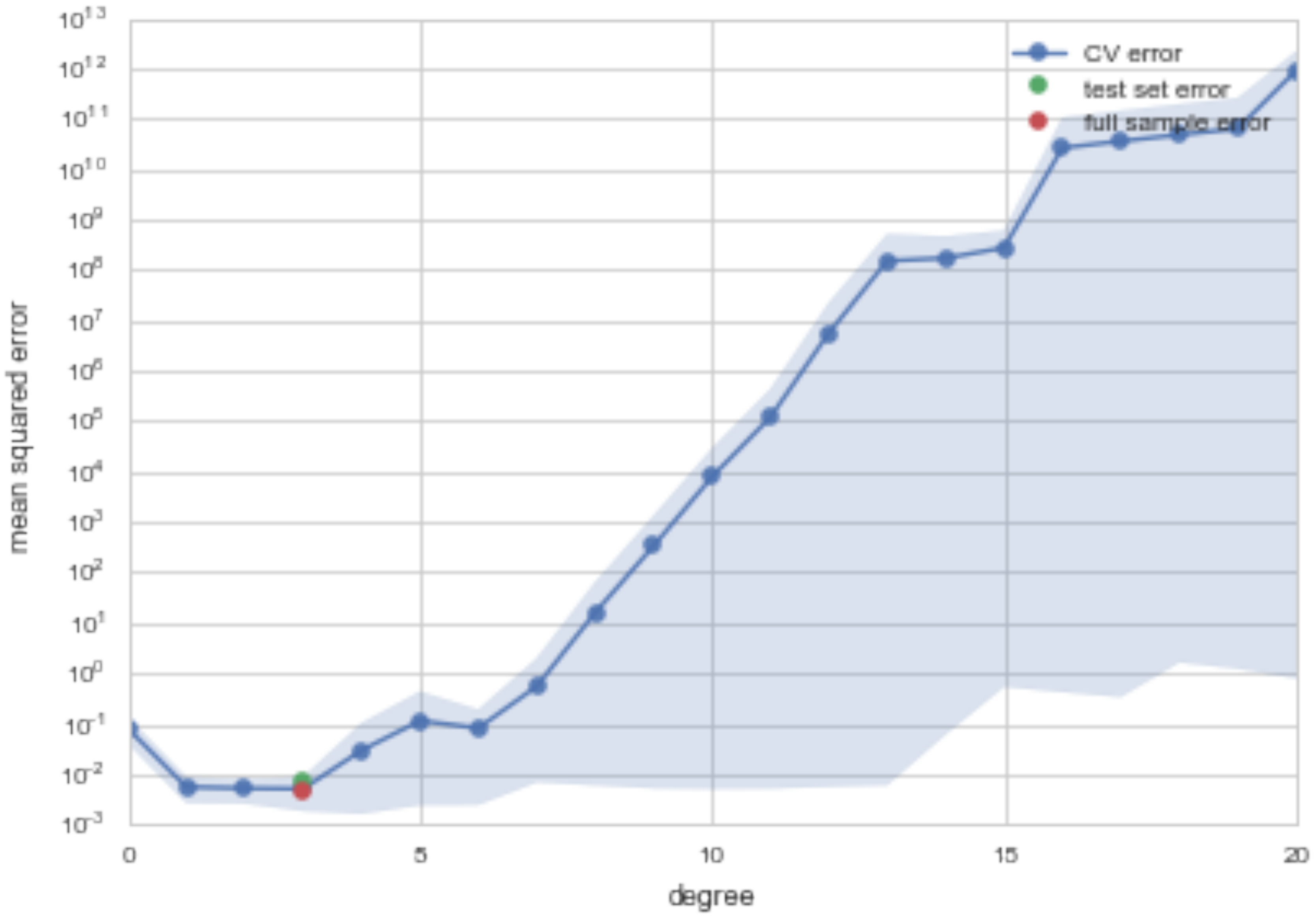

y; fit in $\mathcal{H}_0$
y; fit in $\mathcal{H}_1$
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1.0
−0.5
0.0
0.5
1.0
x

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.

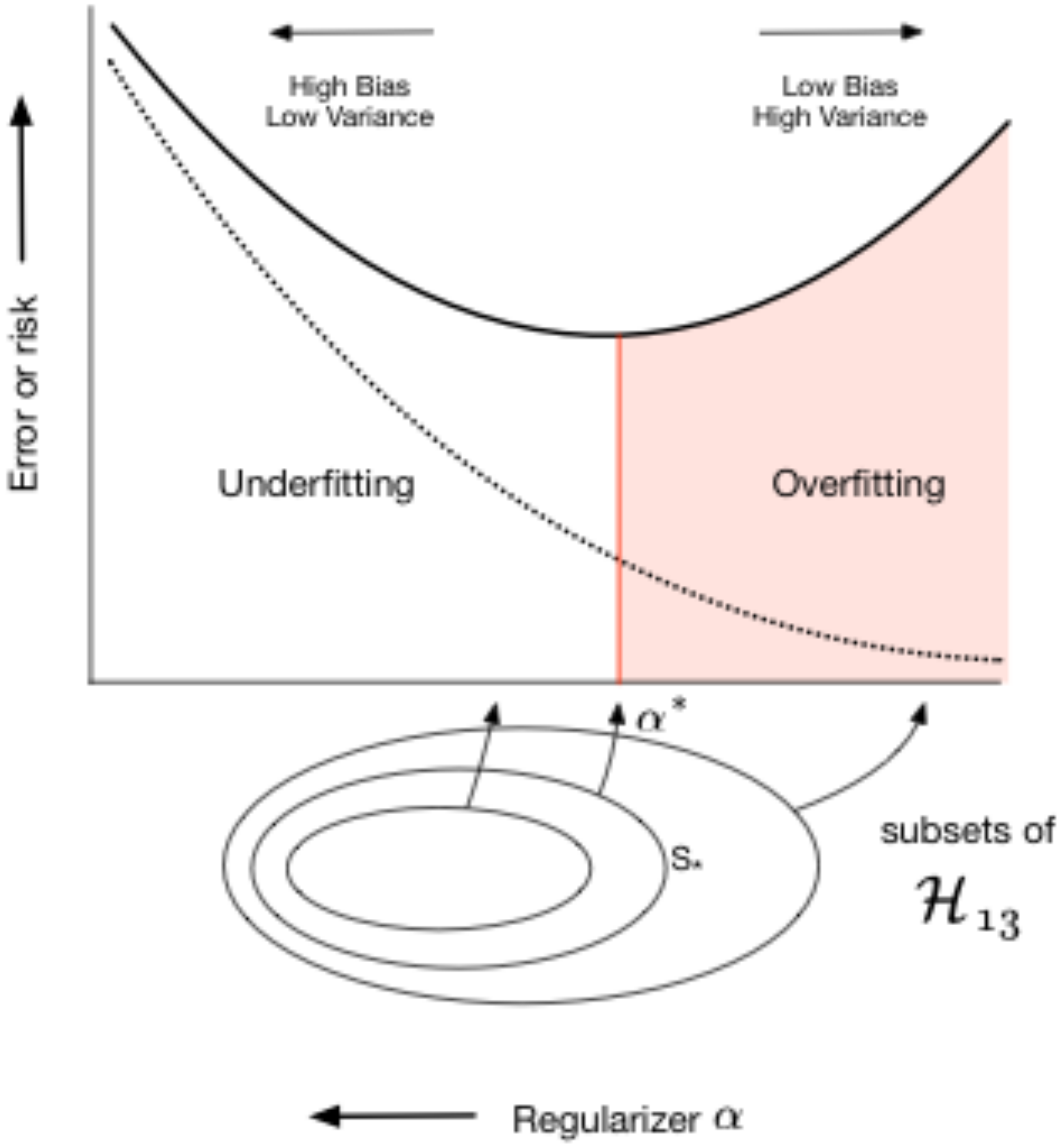
REGULARIZATION

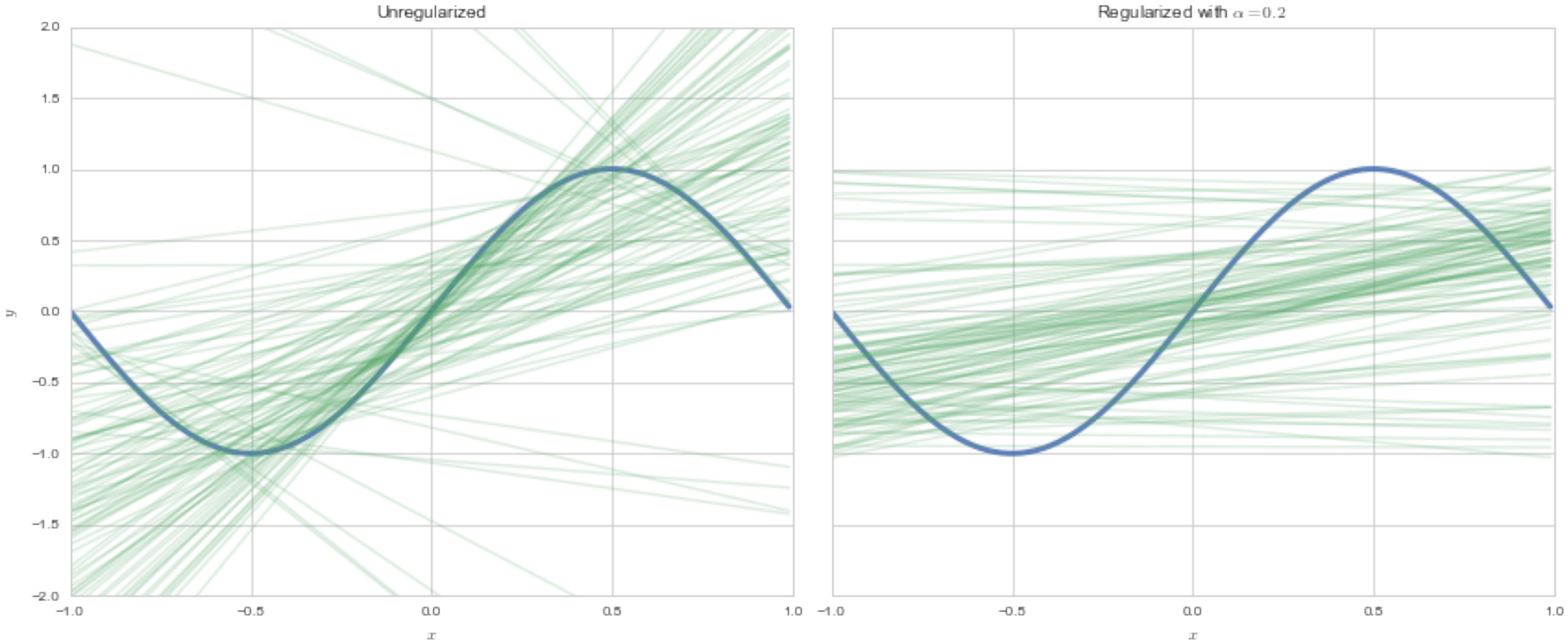
Keep higher a-priori complexity and impose a

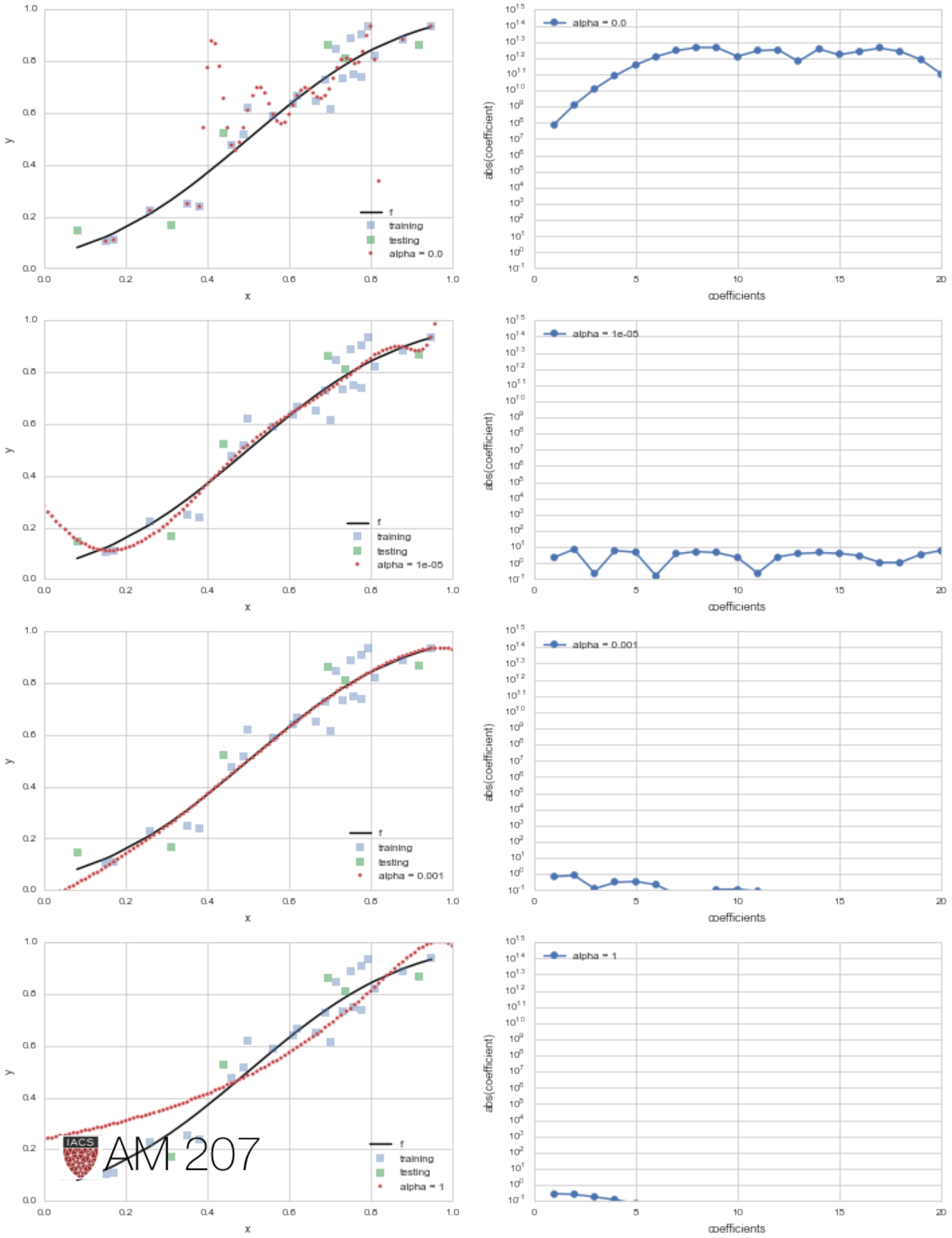
complexity penalty

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
x



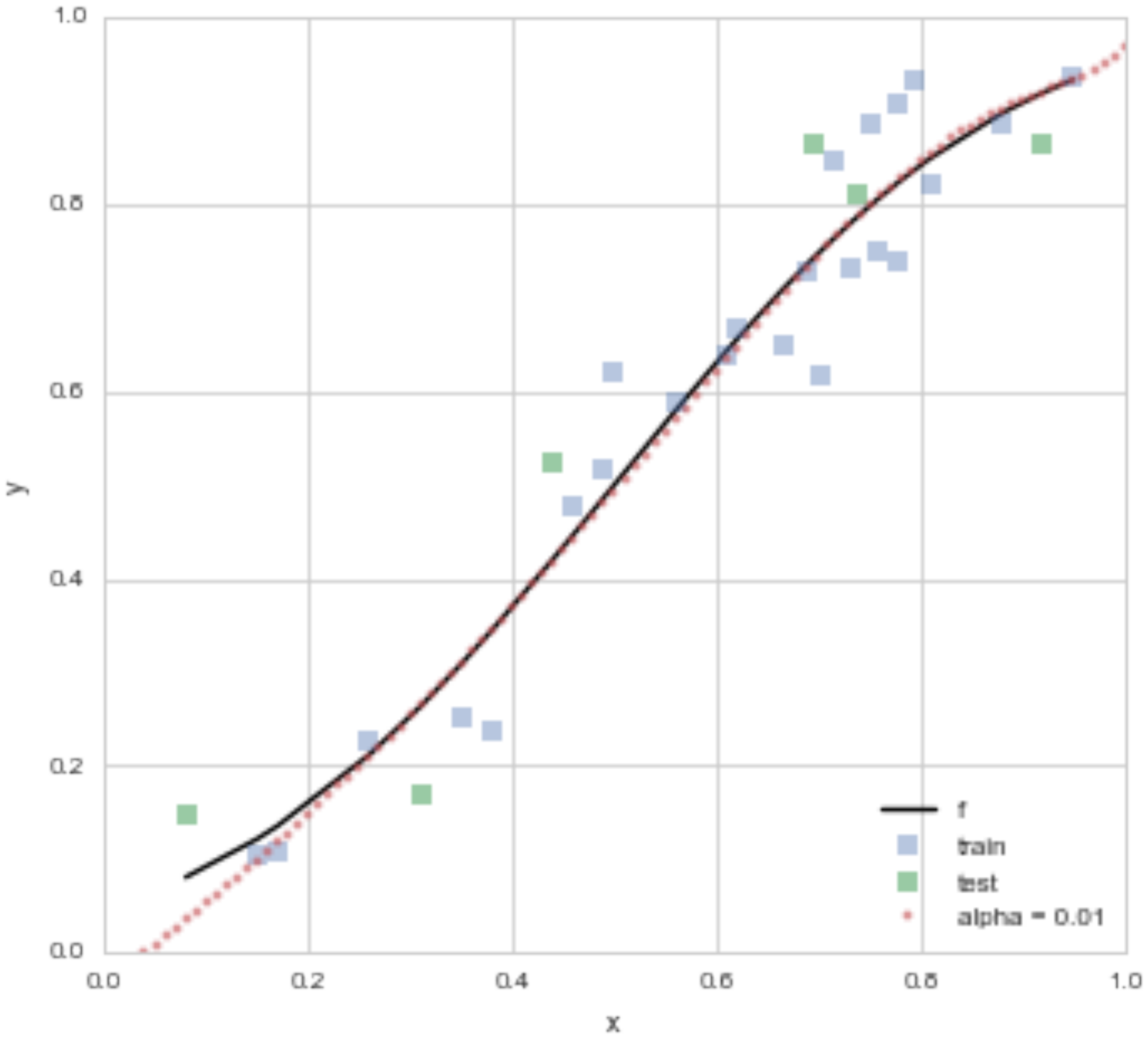
REGULARIZATION

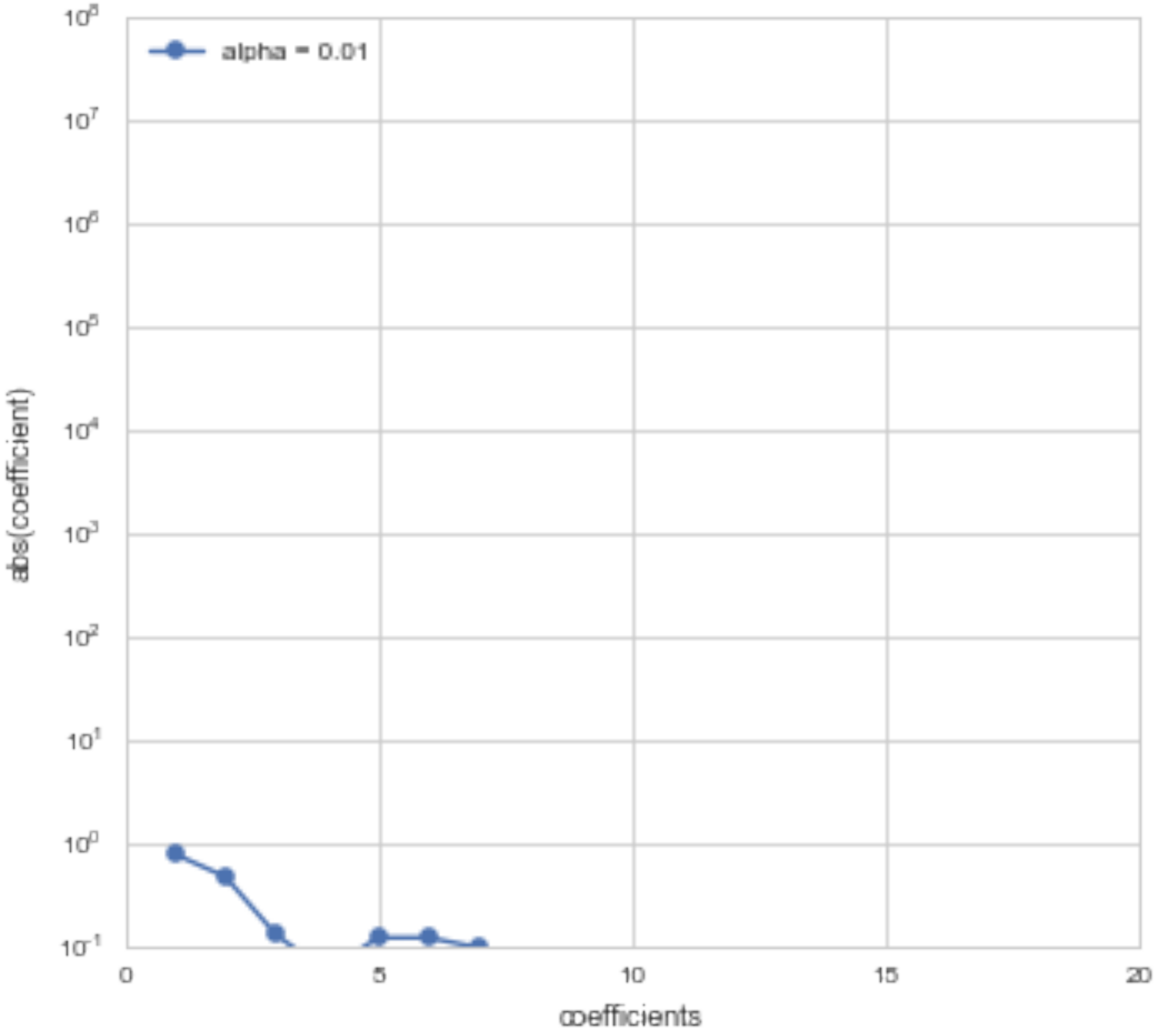
$$\mathcal{R}(h_j) = \sum_{y_i \in \mathcal{D}} (y_i - h_j(x_i))^2 + \alpha \sum_{i=0}^{j} \theta_i^2.$$

As we increase α, coefficients go towards 0.

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

AM 207


1.0
0.8
0.6
0.4
0.2
0.0
y
0.0
0.2
0.4
0.6
0.8
1.0
x
f
train
test
alpha = 0.01


alpha = 0.01
abs(coefficient)
0
5
10
15
20
coefficients

Next time

Minimize the risk

- analytically
- using gradient descent
- using stochastic gradient descent