
Univ.AI

Univ.AI

Learning a Model

Part2

Validation and Regularization

Univ.AI

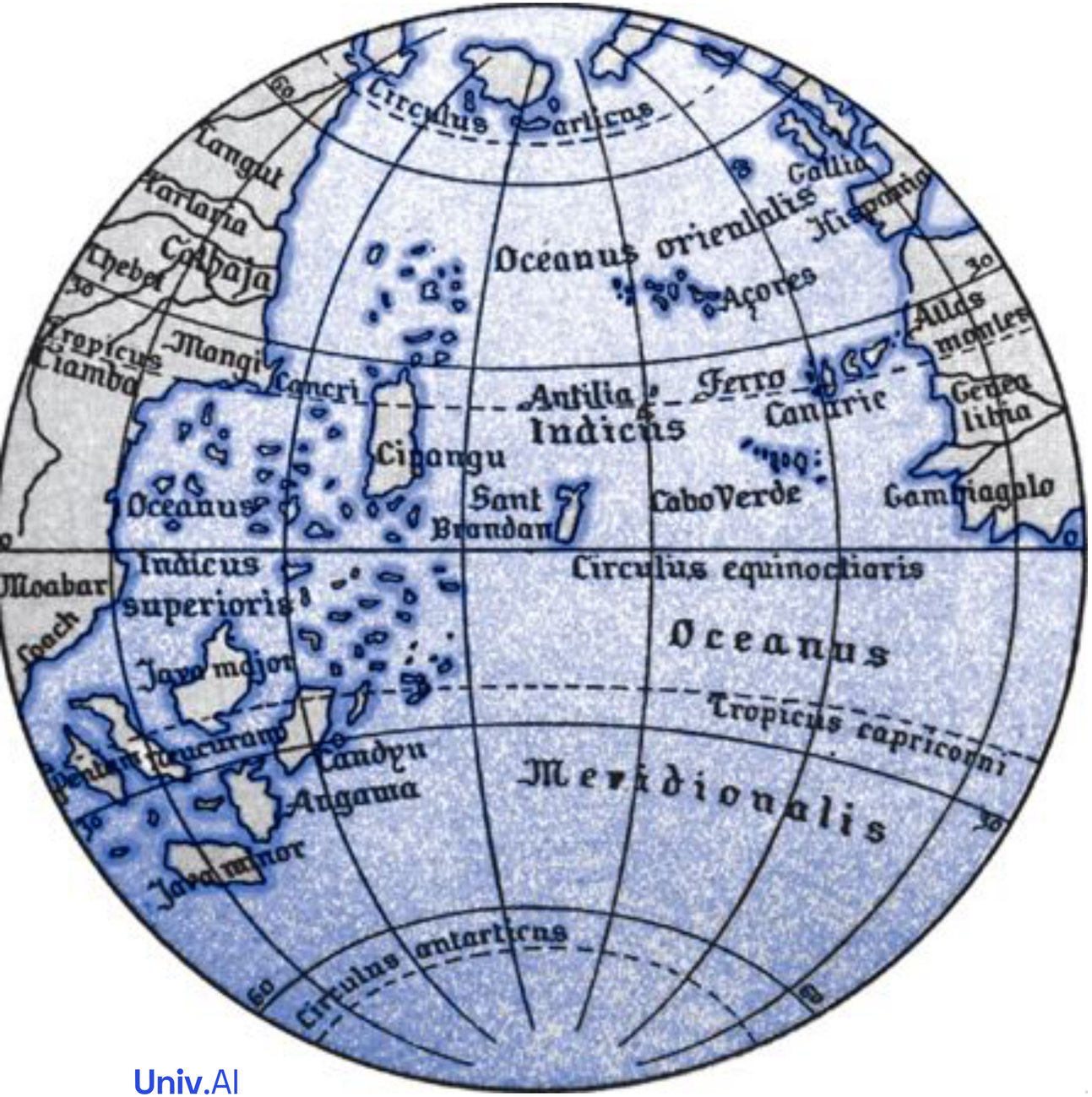
Last Time

1. SMALL World vs BIG World
2. Approximation
3. THE REAL WORLD HAS NOISE
4. Complexity amongst Models
5. Validation

Today

- (0) Recap of key concepts from earlier
- (1) Validation and Cross Validation
- (2) Regularization
- (3) Multiple Features

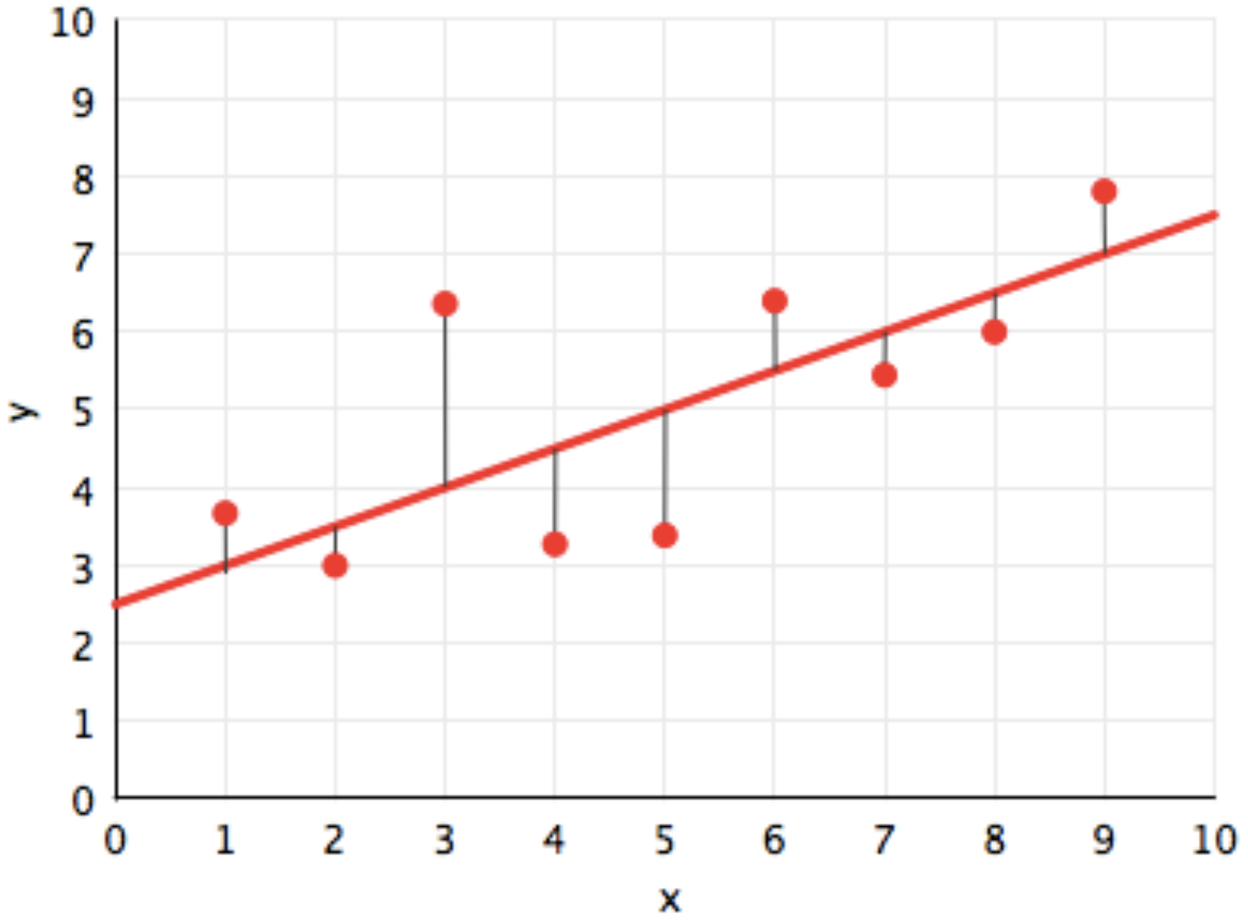
0. From before



- *Small World* given a map or model of the world, how do we do things in this map?
- *BIG World* compares maps or models. Asks: whats the best map?

(Behaim Globe, 21 inches (51 cm) in diameter and was fashioned from a type of papier-mache and coated with gypsum. (wikipedia))

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.

$$g_1(x) = \arg\min_{h_1(x) \in \mathcal{H}_1} R_{\mathcal{D}}(h_1(x)).$$

Get intercept w_0 and slope w_1.

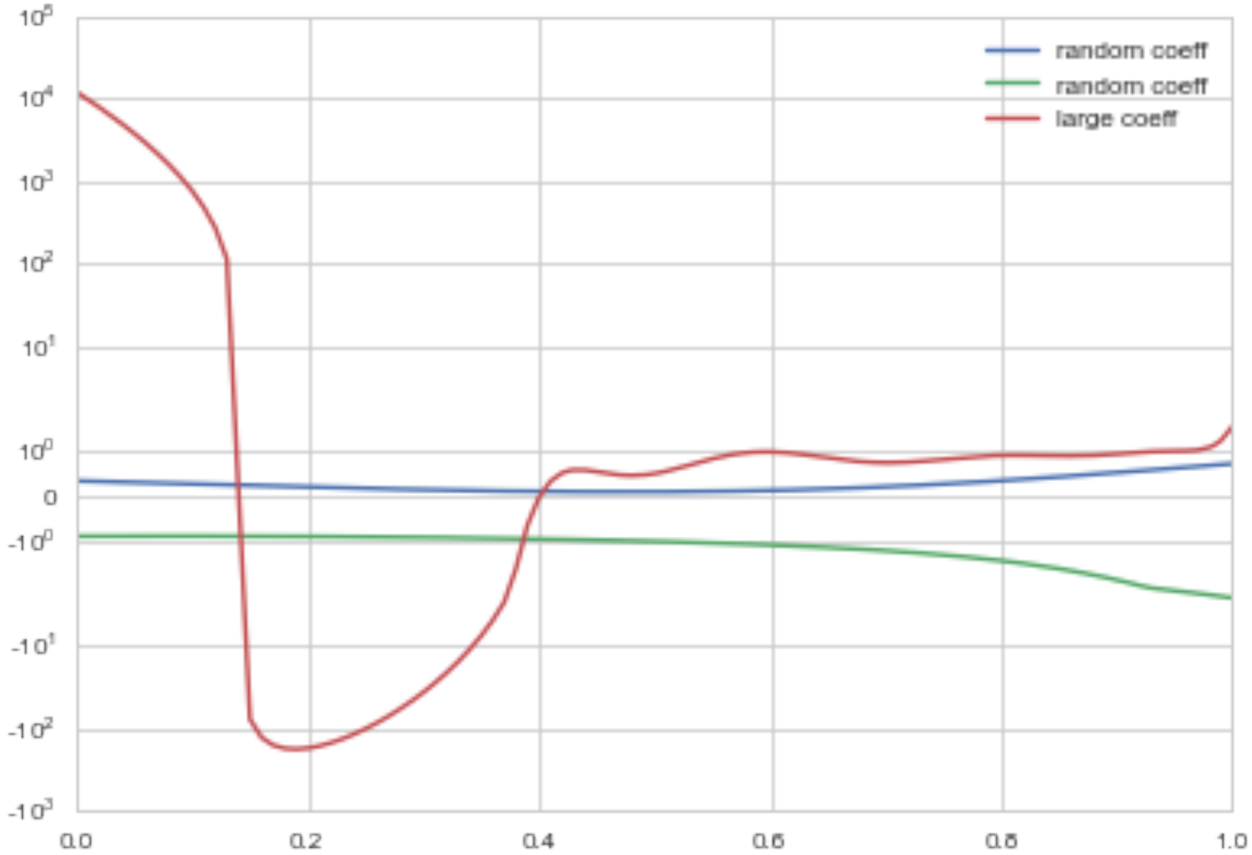
HYPOTHESIS SPACES

For example, a polynomial looks so:

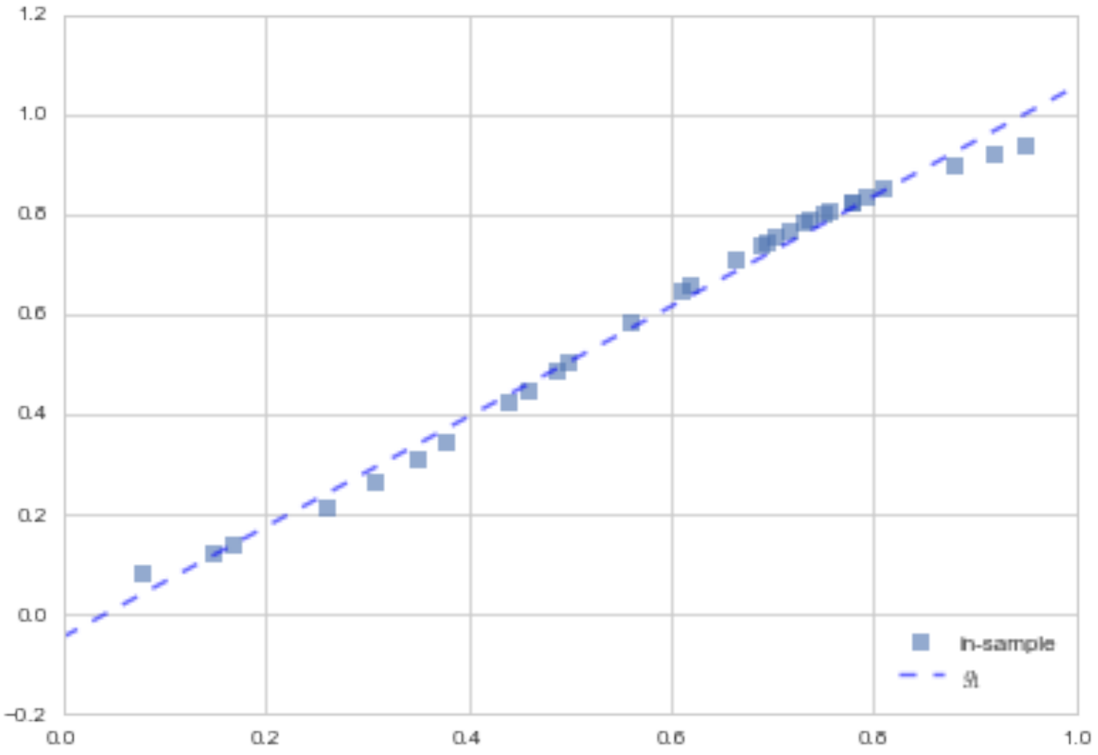
$$h(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2 + \ldots + \theta_n x^n = \sum_{i=0}^{n} \theta_i x^i$$

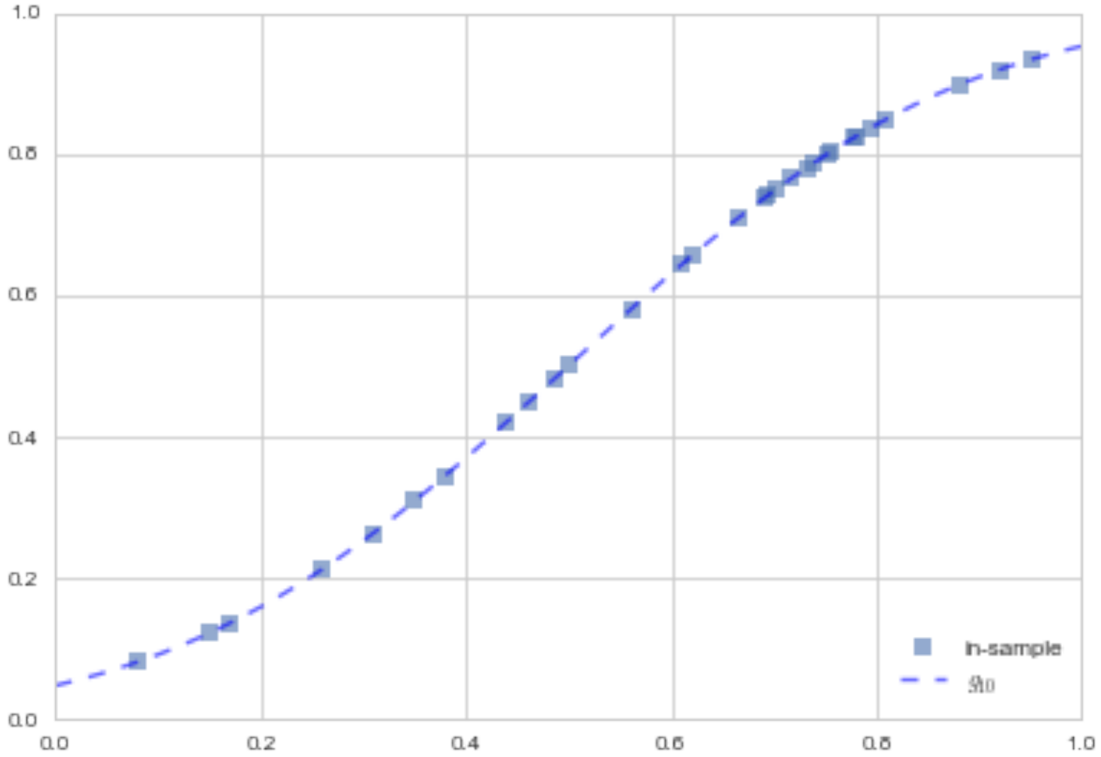
All polynomials of a degree or complexity d constitute a hypothesis space.

$$\mathcal{H}_1 : h_1(x) = \theta_0 + \theta_1 x$$
$$\mathcal{H}_{20} : h_{20}(x) = \sum_{i=0}^{20} \theta_i x^i$$

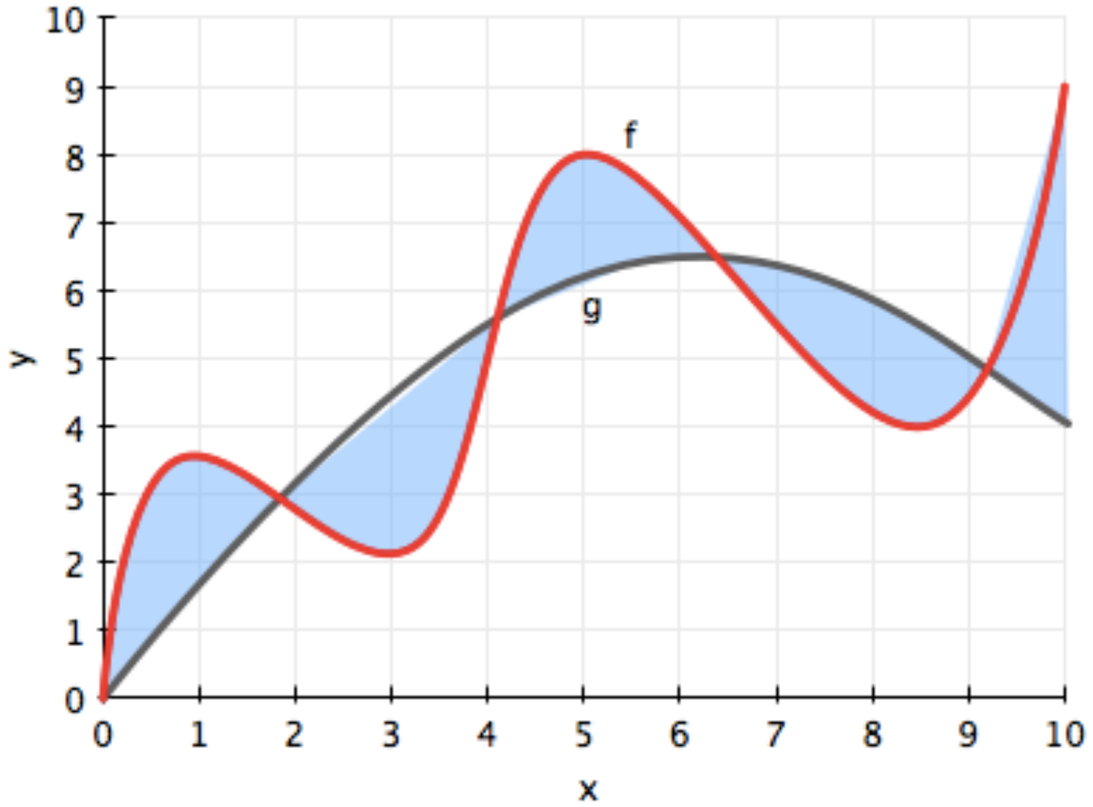


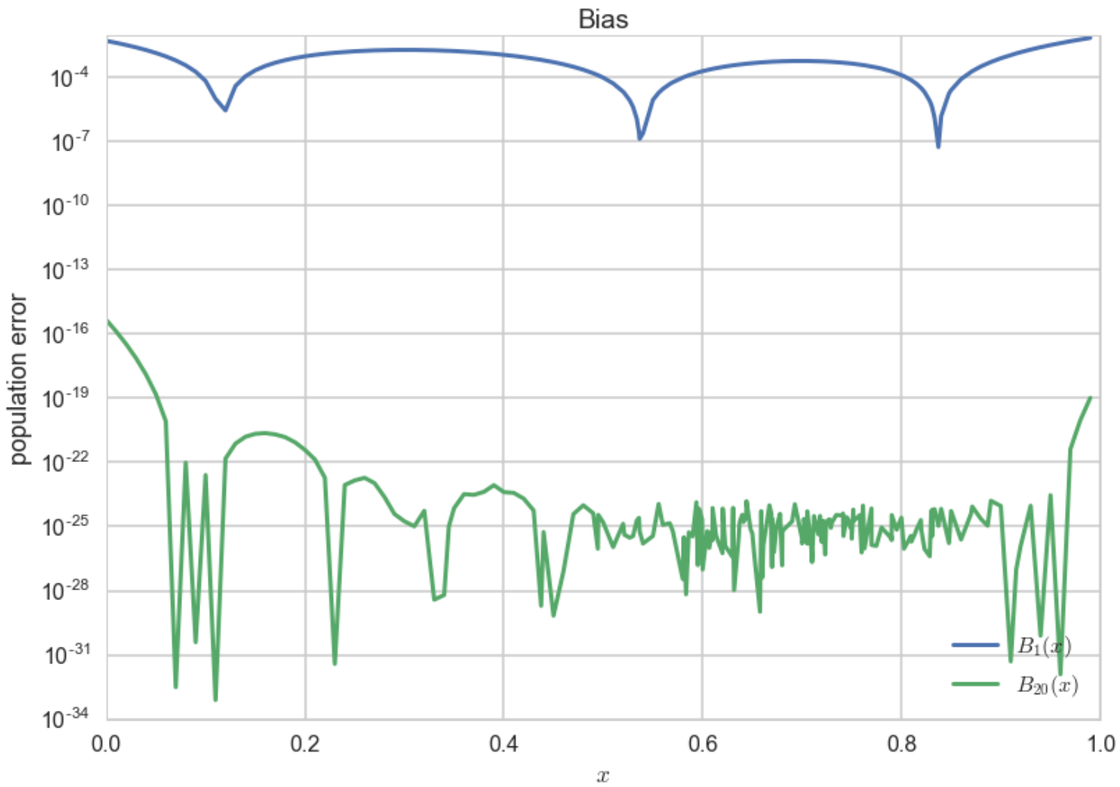
A sample of 30 points of data. Which fit is better? Line in $\mathcal{H}_1$ or curve in $\mathcal{H}_{20}$?

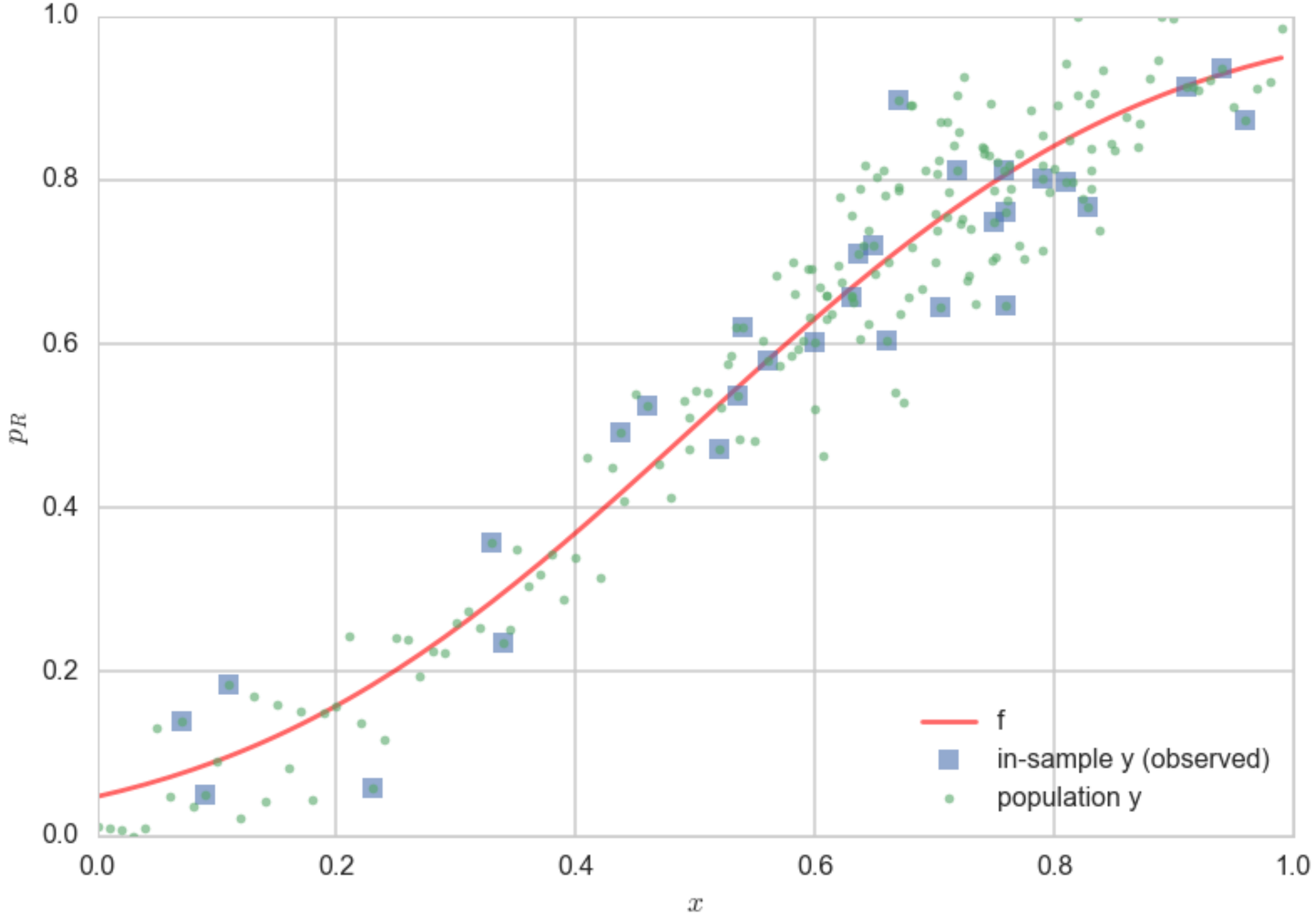


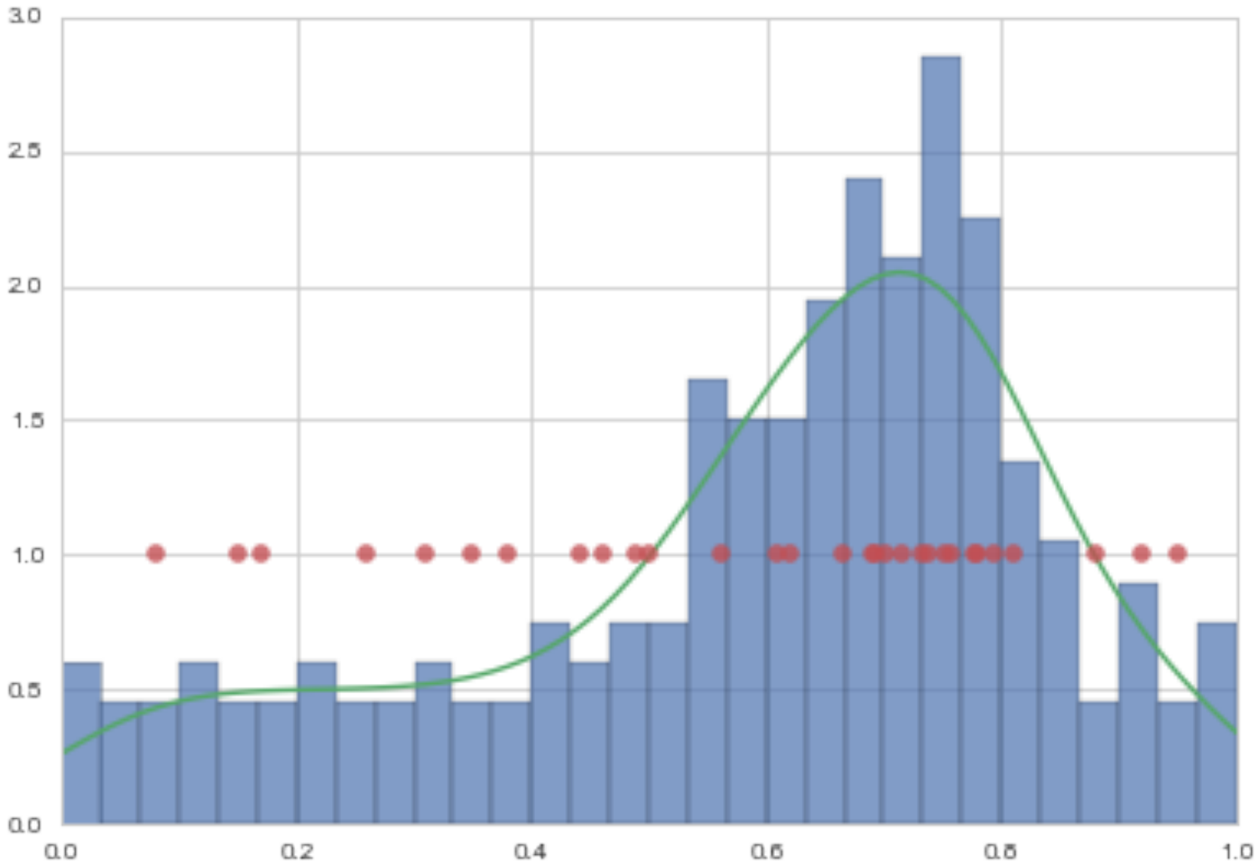


Bias or Mis-specification Error






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



Statement of the Learning Problem

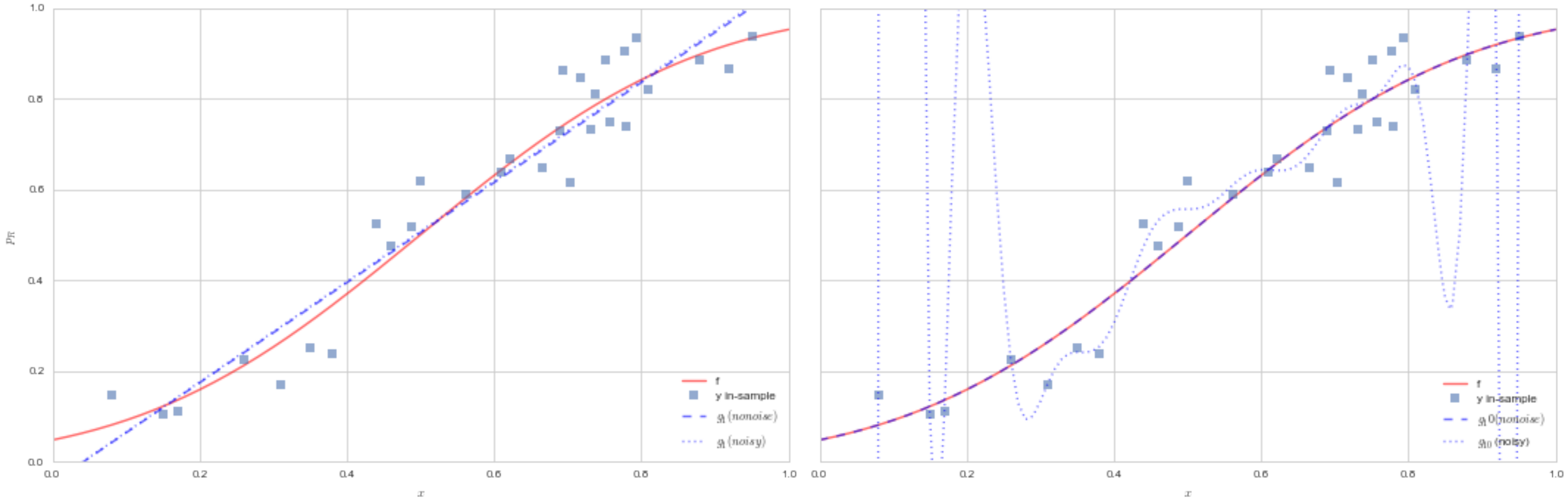
The sample must be representative of the population!

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

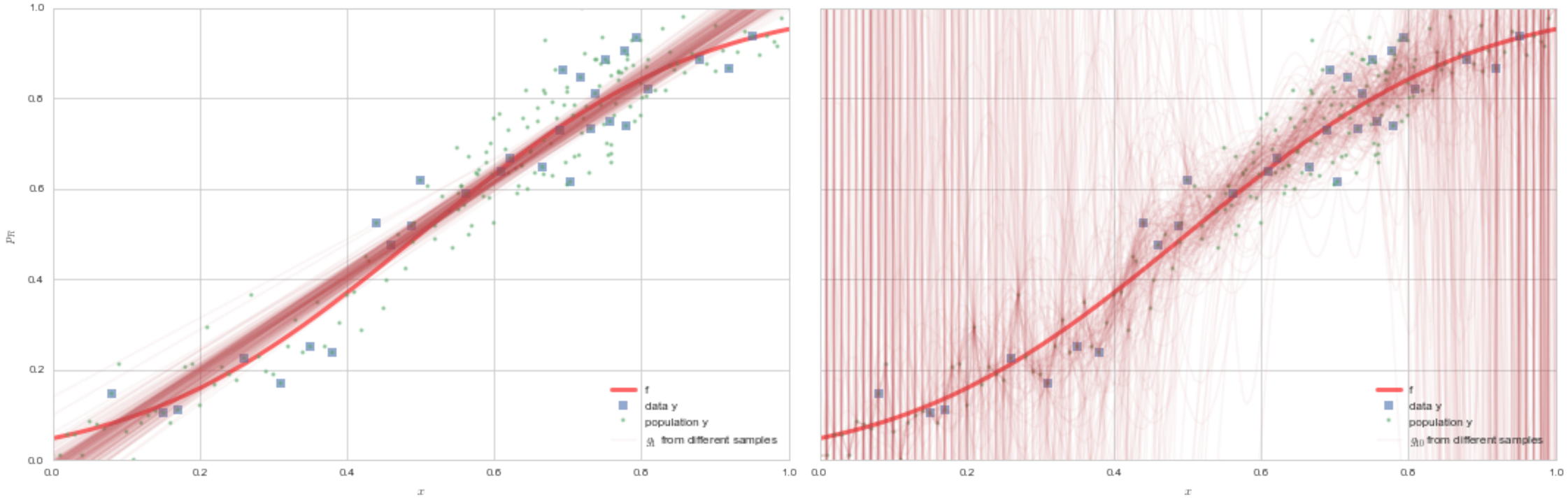
A: In-sample risk is small
B: Population, or out-of-sample risk is WELL estimated by in-sample risk. Thus the out of sample risk is also small.

Which fit is better now?

The line or the curve?



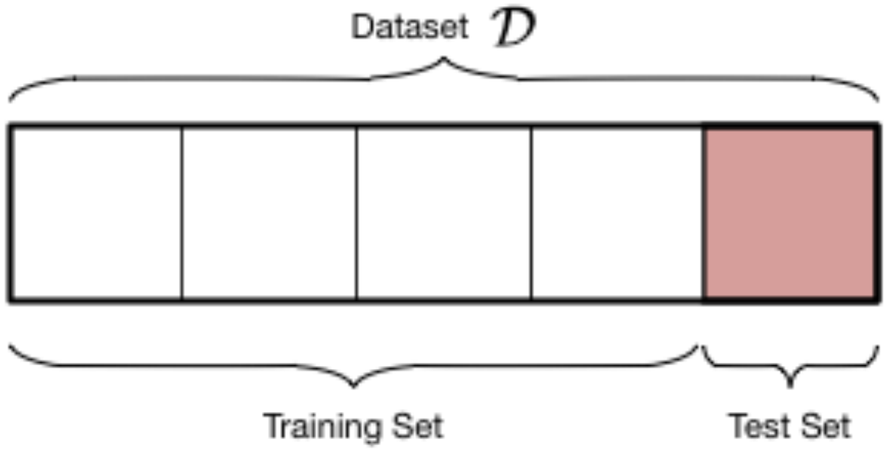
UNDERFITTING (Bias) vs OVERFITTING (Variance)

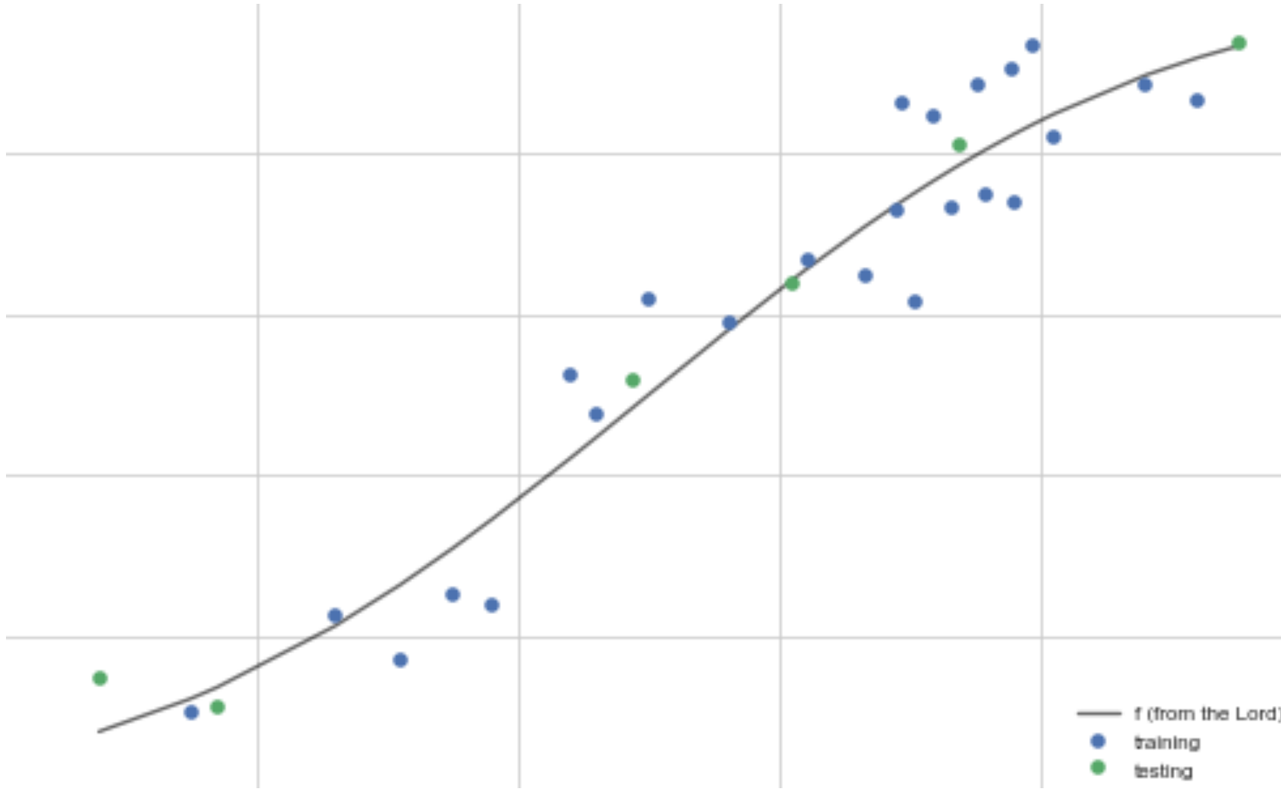


How do we estimate

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

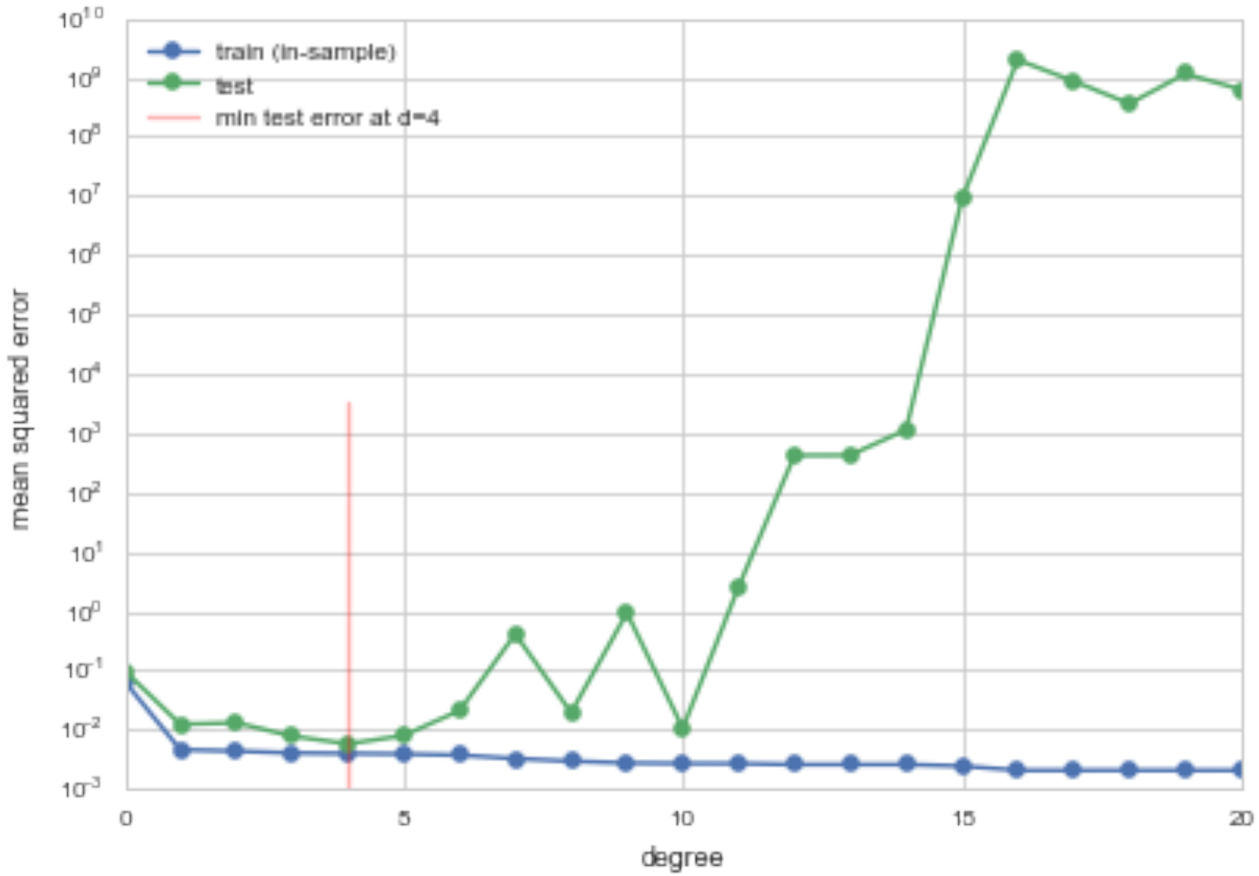
TRAIN AND TEST



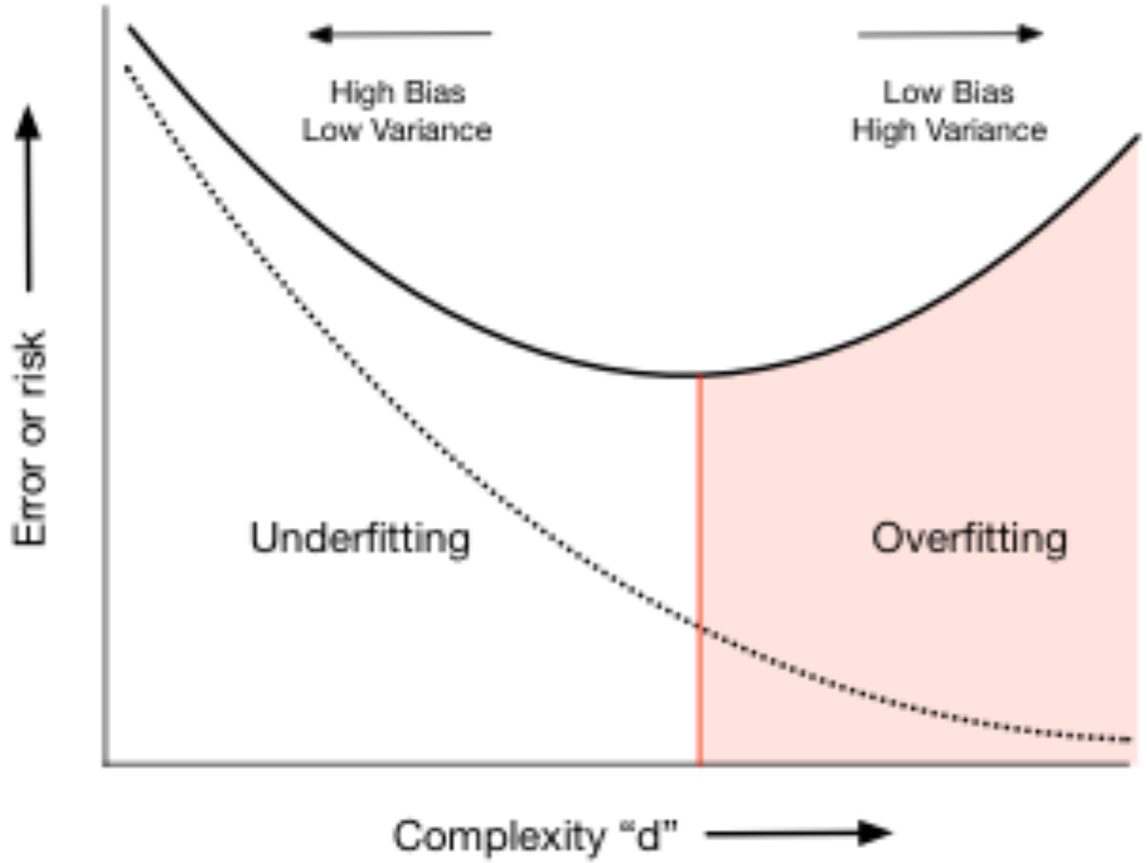


MODEL COMPARISON: A Large World approach

- 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



1. Validation and Cross Validation

Do we still have a test set?

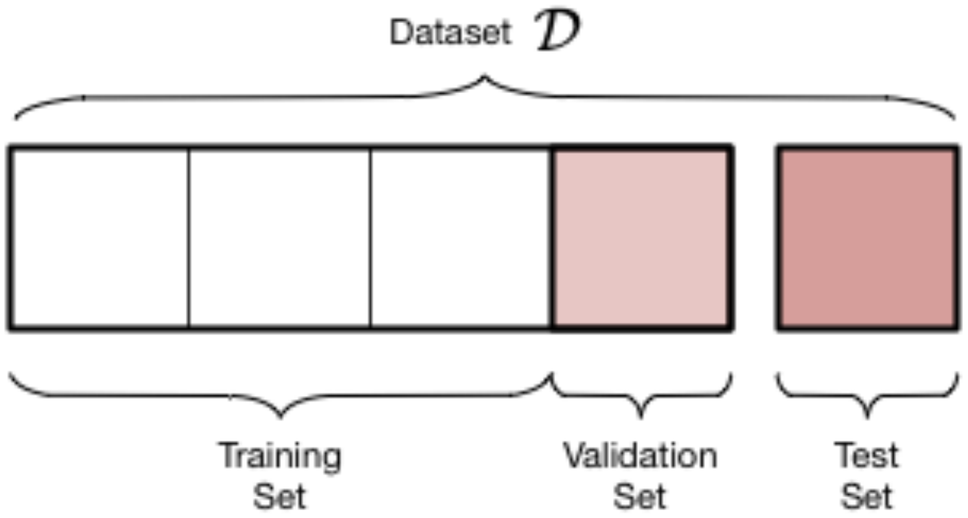
Trouble:

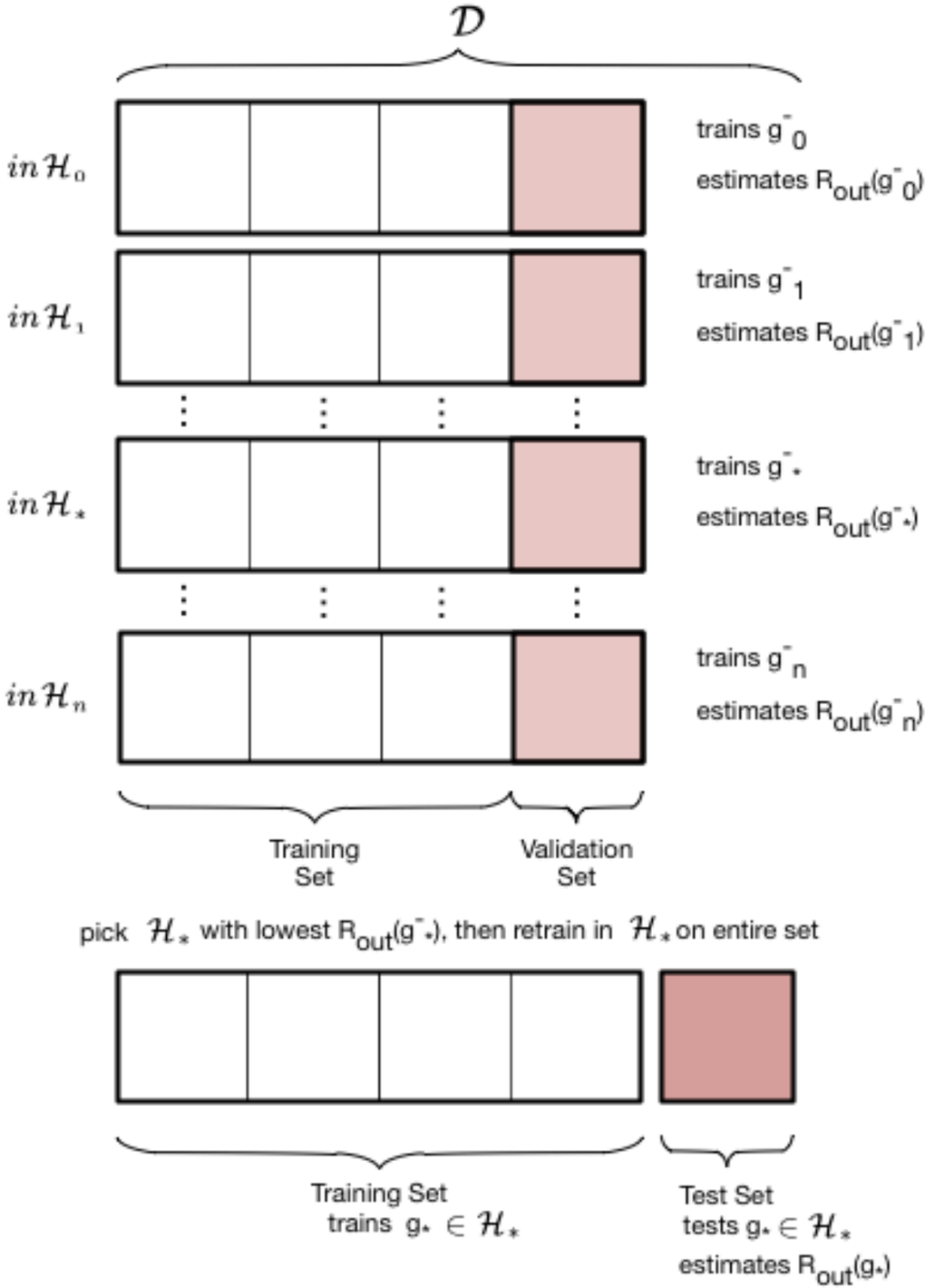
- no discussion on the error bars on our error estimates
- "visually fitting" a value of $d \implies$ contaminated test set.

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.

Whats the problem?

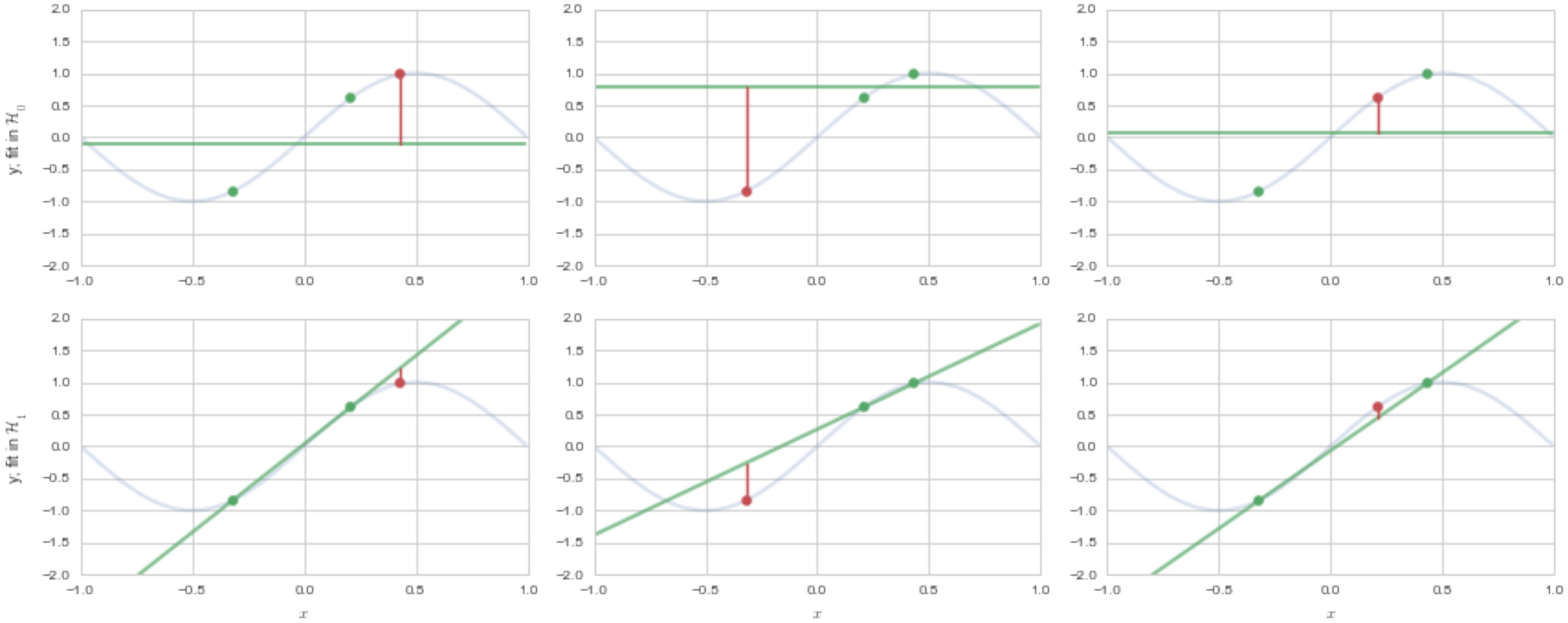
What if we, just by chance had an iffy validation set?

This problem is dire when we are in low data situations. In large data situations, not so much.

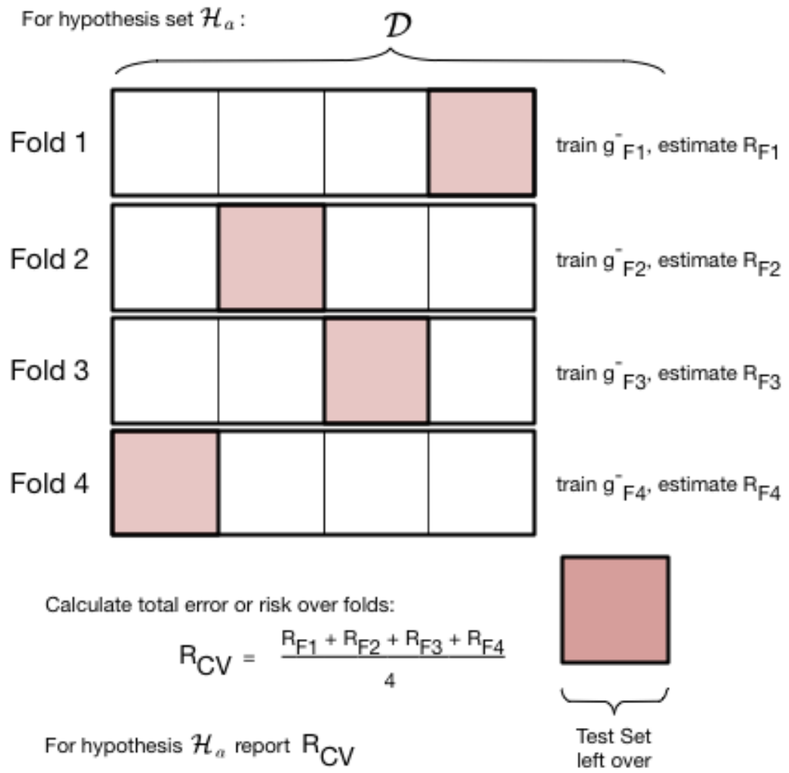
We then do

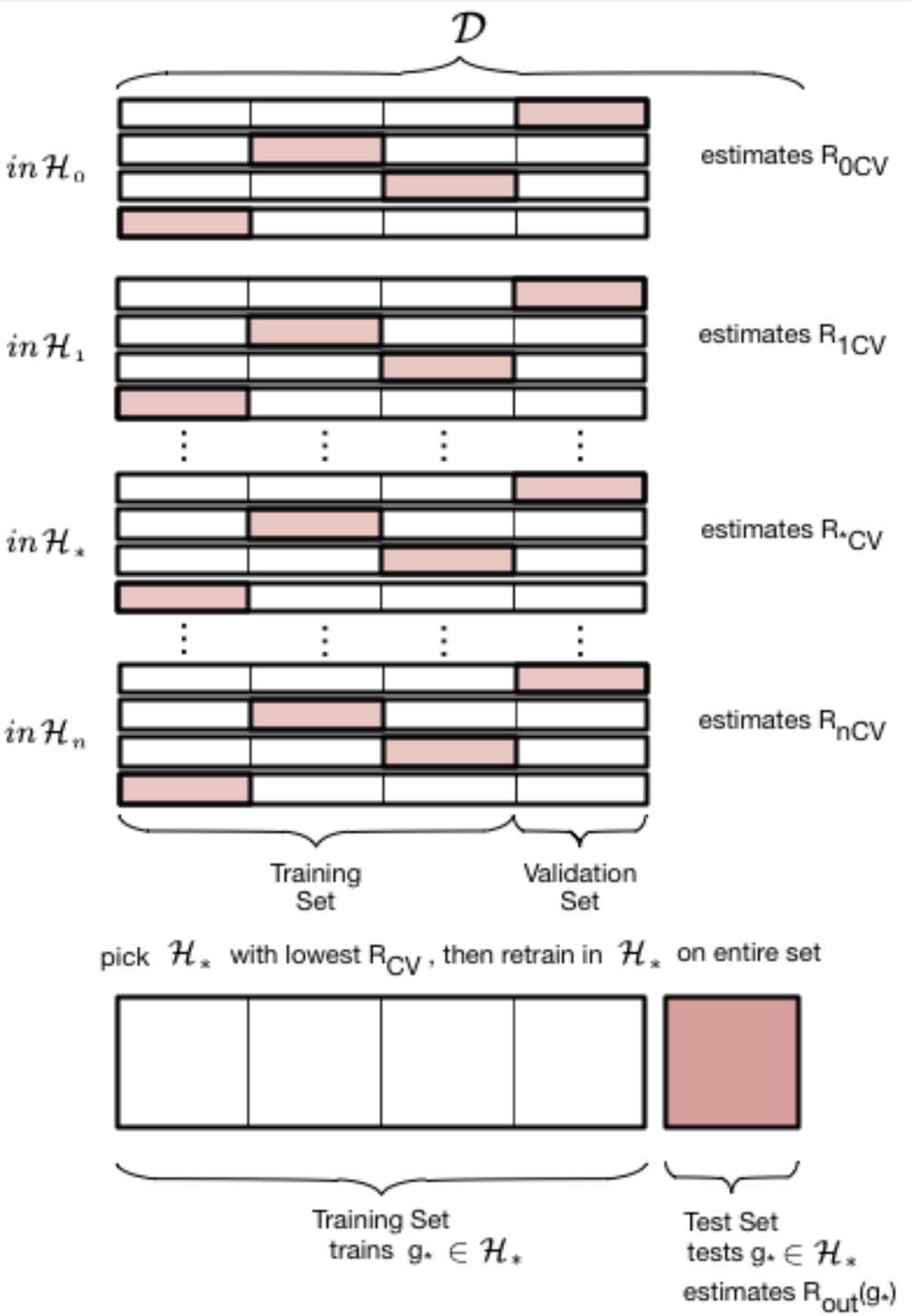
cross-validation

Key Idea: Repeat the validation process on different pieces of left out data. Make these left-out parts not overlap so that the risks/errors/mse calculated on each are not correlated.


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



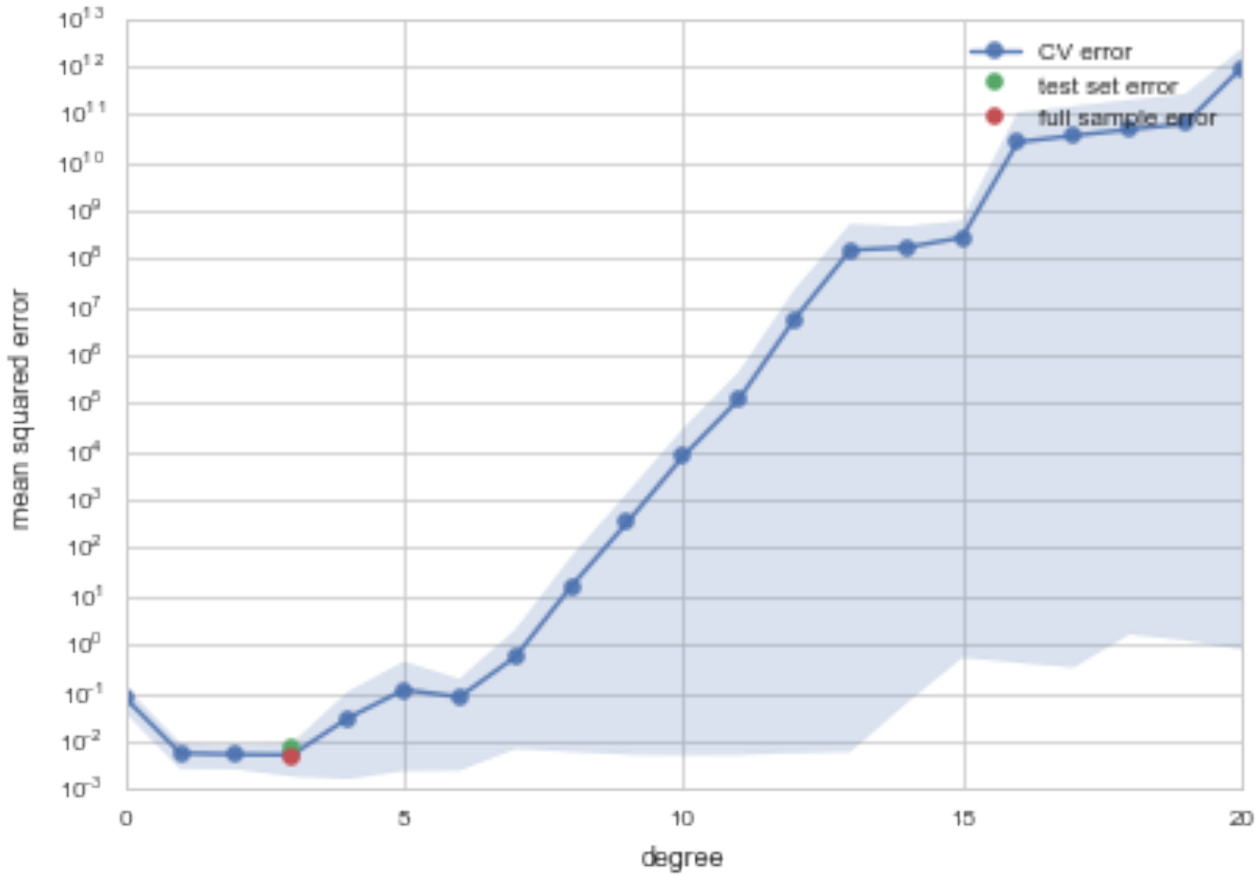

$\mathcal{D}$
$in\,\mathcal{H}_0$
estimates R_{0CV}
$in\,\mathcal{H}_1$
estimates R_{1CV}
$in\,\mathcal{H}_*$
estimates R_{*CV}
$in\,\mathcal{H}_n$
estimates R_{nCV}
Training Set
Validation Set
pick $\mathcal{H}_*$ with lowest R_{CV}, then retrain in $\mathcal{H}_*$ on entire set
Training Set
trains $g_* \in \mathcal{H}_*$
Test Set
tests $g_* \in \mathcal{H}_*$
estimates $R_{out}(g_*)$

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.

Consider a "small-world" approach to deal with finding the right model, where we'll choose a Hypothesis set that includes very complex models, and then find a way to subset this set.

This method is called

2. Regularization

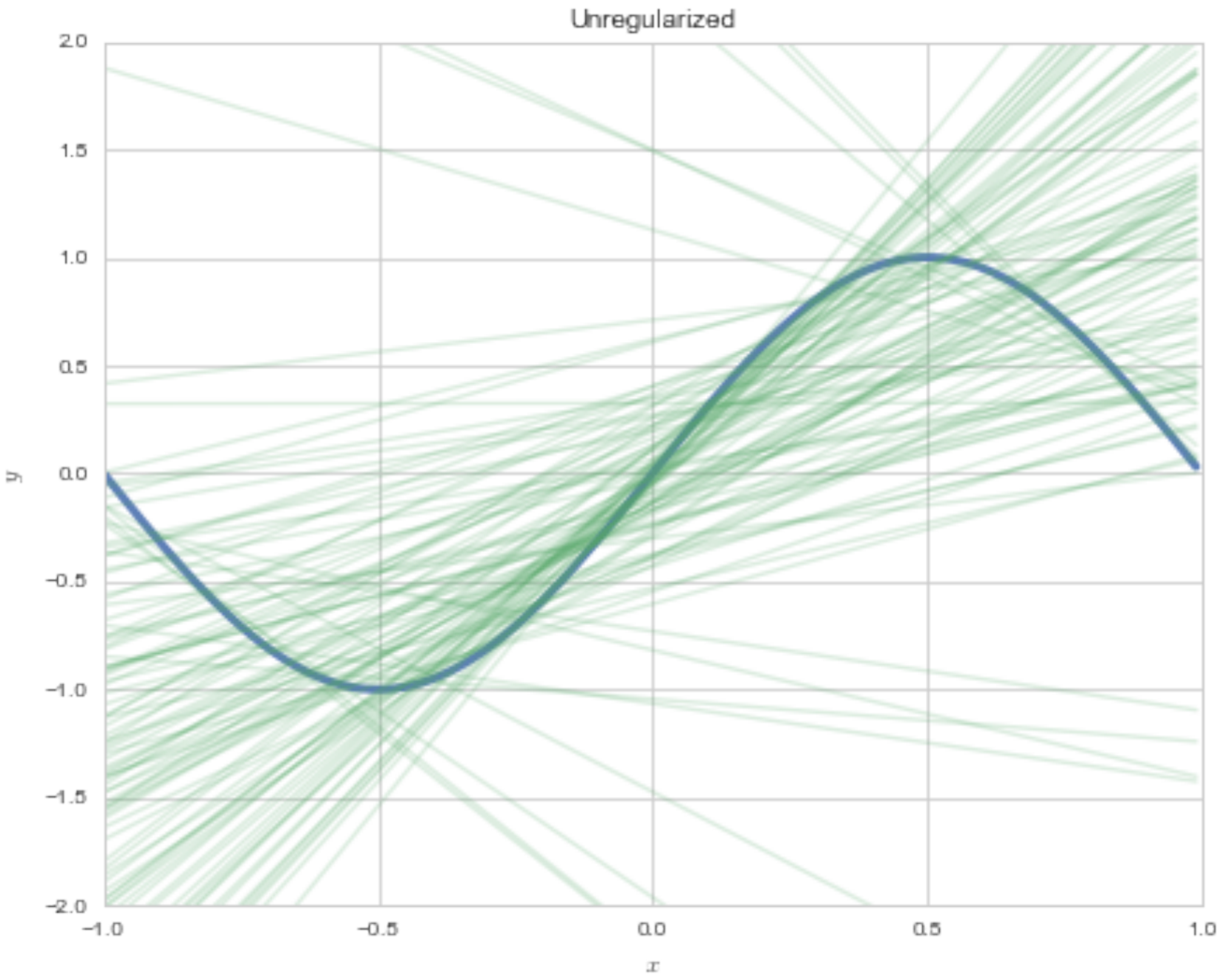
REGULARIZATION: A SMALL WORLD APPROACH

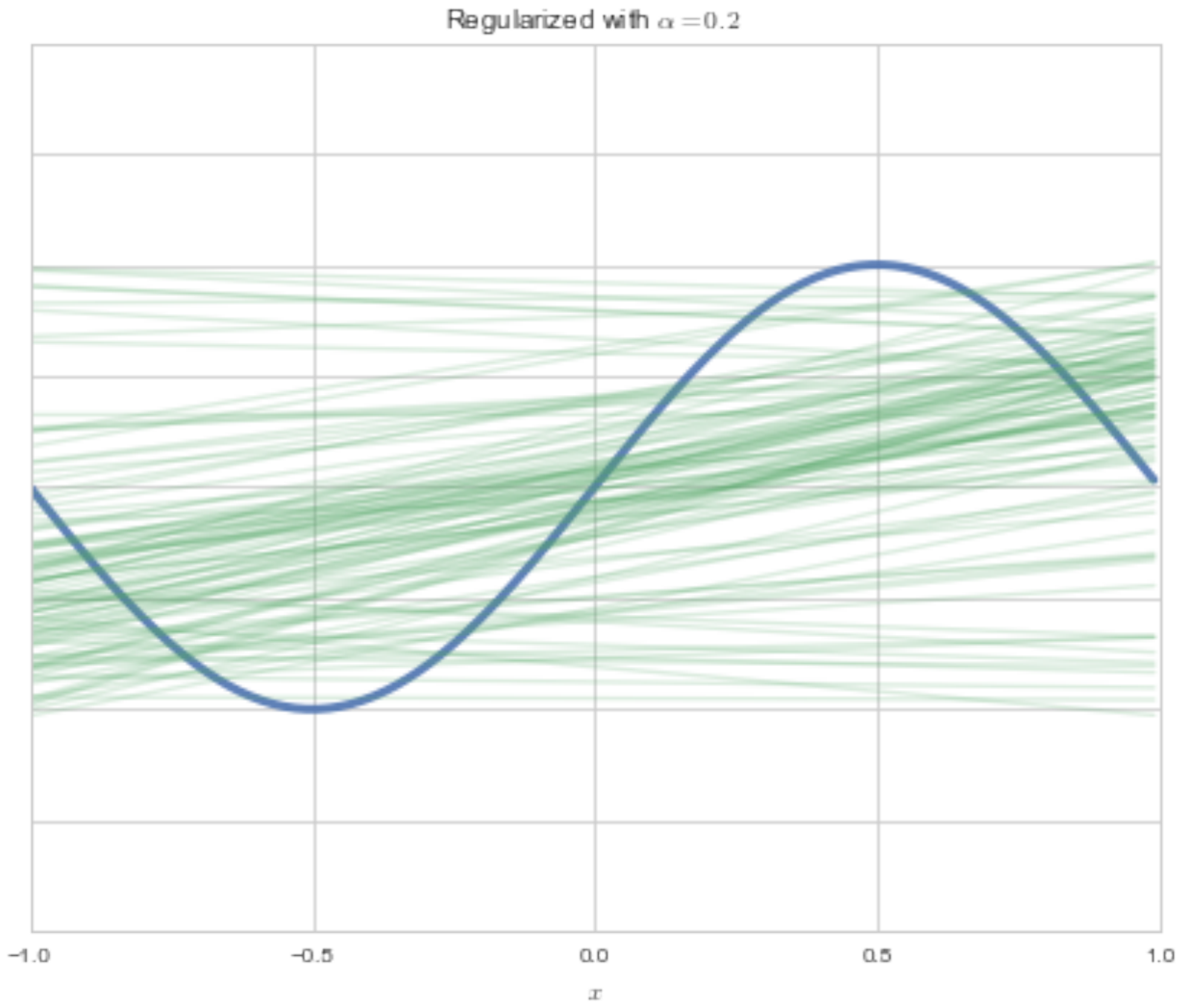
Keep higher a-priori complexity and impose a

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
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


Regularized with $\alpha = 0.2$
−1.0
−0.5
0.0
0.5
1.0
x

The math of regularization: small world

Consider the set of 10th order polynomials:

$$\mathcal{H}_{10} = \{h(x) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + w_3\Phi_3(x) + \cdots + w_{10}\Phi_{10}(x)\}$$

Now suppose we just set some of these to 0, then we get $\mathcal{H}_2$ as a subset:

$$\mathcal{H}_2 = \begin{cases} h(x) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + w_3\Phi_3(x) + \cdots + w_{10}\Phi_{10}(x) \\ \textbf{such that: } w_3 = w_4 = \cdots = w_{10} = 0 \end{cases}$$

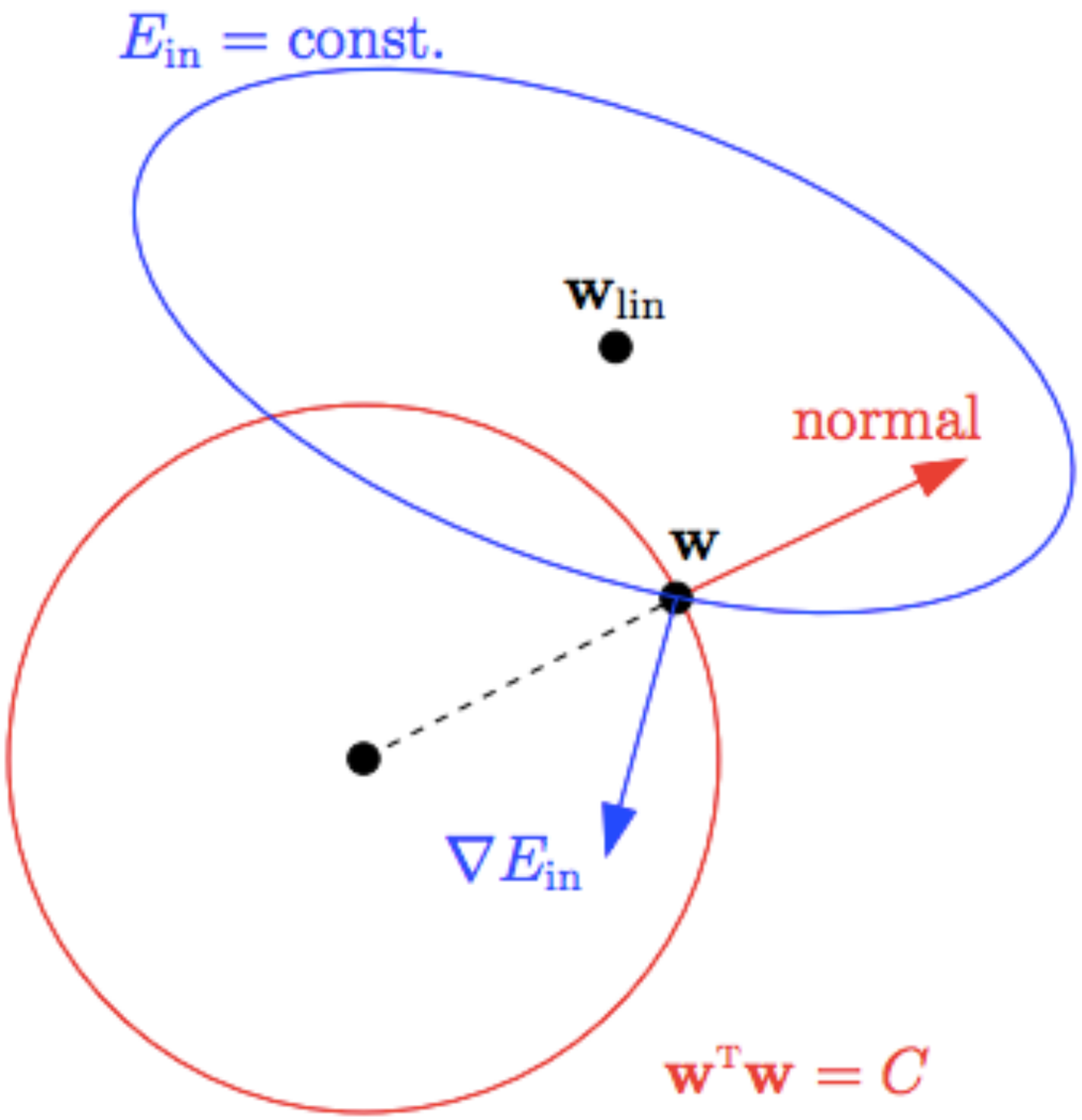
This is called a hard-order constraint.

L_2 Regularization or a soft budget constraint

$$\sum_{q=0}^{\mathrm{Q}} w_q^2 \leq C \leftarrow \text{BUDGET}$$

$$\mathcal{H}_C = \begin{cases} h(x) = w_0 + w_1\Phi_1(x) + w_2\Phi_2(x) + w_3\Phi_3(x) + \cdots + w_{10}\Phi_{10}(x) \\ \text{such that: } \sum_{q=0}^{10} w_q^2 \leq C \end{cases}$$

a soft budget constraint



The geometry of regularization

1. Optimal $\mathbf{w}$ tries to get as 'close' to $\mathbf{w}_{lin}$. Thus, optimal $\mathbf{w}$ will use full budget and be on the surface $\mathbf{w}^T\mathbf{w} = C$.
2. Surface $\mathbf{w}^T\mathbf{w} = C$, at optimal $\mathbf{w}$, should be perpindicular to ∇E_{in}.
3. Normal to surface $\mathbf{w}^T\mathbf{w} = C$ is the vector $\mathbf{w}$.
4. Surface is $\perp \nabla E_{\text{in}}$ and thus must be "tangent"

$$\nabla E_{\text{in}} \left(\mathbf{w}_{\text{reg}}\right) = -2\lambda_C \mathbf{w}_{\text{reg}}$$

Back to the Math: the lagrange multiplier formalism

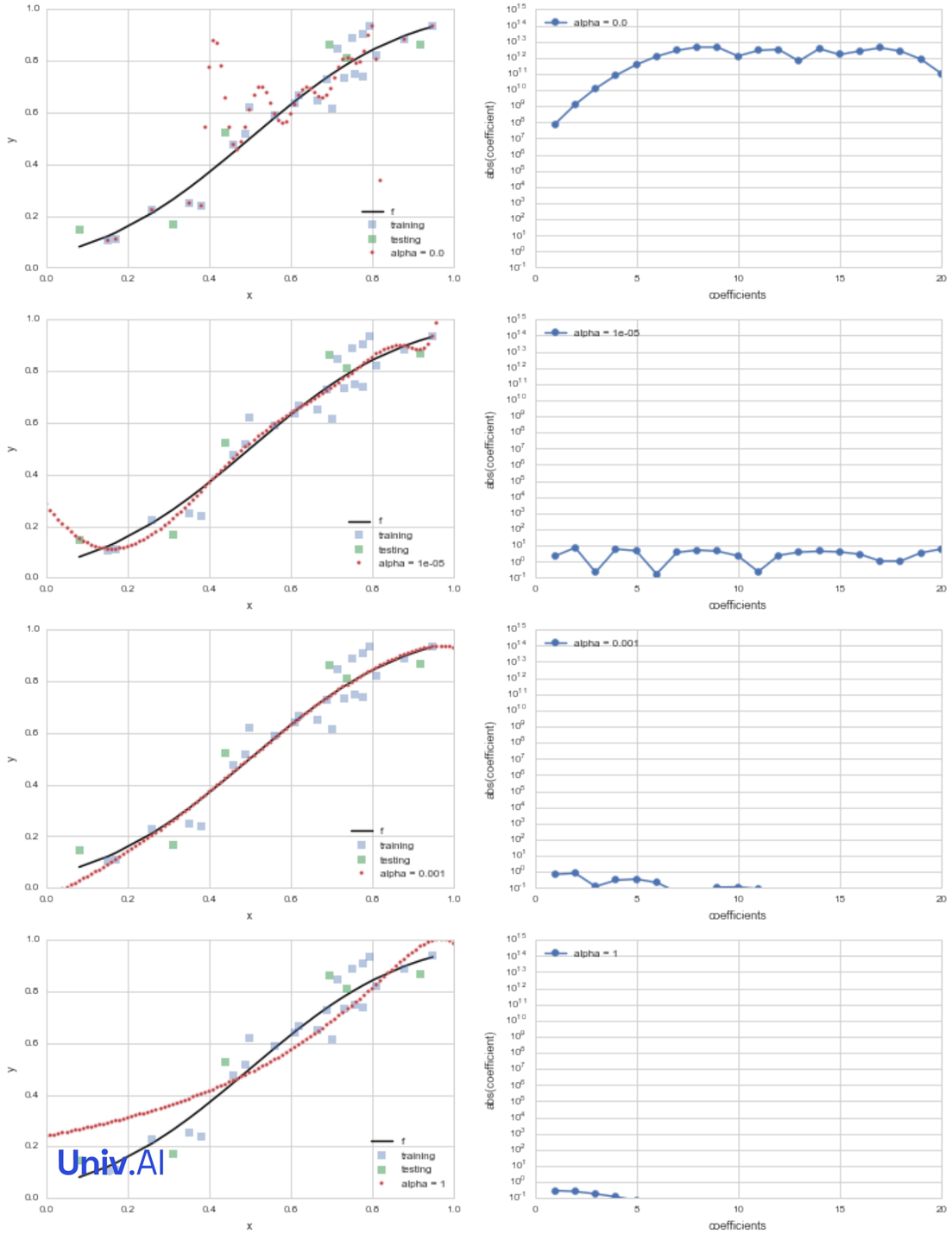
$$E_{\text{in}}(\mathbf{w}) \quad \text{is minimized, subject to: } \mathbf{w}^{\mathrm{T}}\mathbf{w} \leq C$$

$$\Leftrightarrow \quad \nabla E_{\text{in}}\left(\mathbf{w}_{\text{reg}}\right) + 2\lambda_C \mathbf{w}_{\text{reg}} = \mathbf{0}$$

$$\Leftrightarrow \nabla \left(E_{\text{in}}(\mathbf{w}) + \lambda_C \mathbf{w}^{\mathrm{T}}\mathbf{w}\right)\Big|_{\mathbf{w}=\mathbf{w}_{\text{rgg}}} = \mathbf{0}$$

$$\Leftrightarrow \quad E_{\text{in}}(\mathbf{w}) + \lambda_C \mathbf{w}^{\mathrm{T}}\mathbf{w} \quad \text{is minimized, unconditionally}$$

$$\text{There is a correspondence: } C \uparrow \quad \lambda_C \downarrow$$



ok so now how do we do

REGULARIZATION

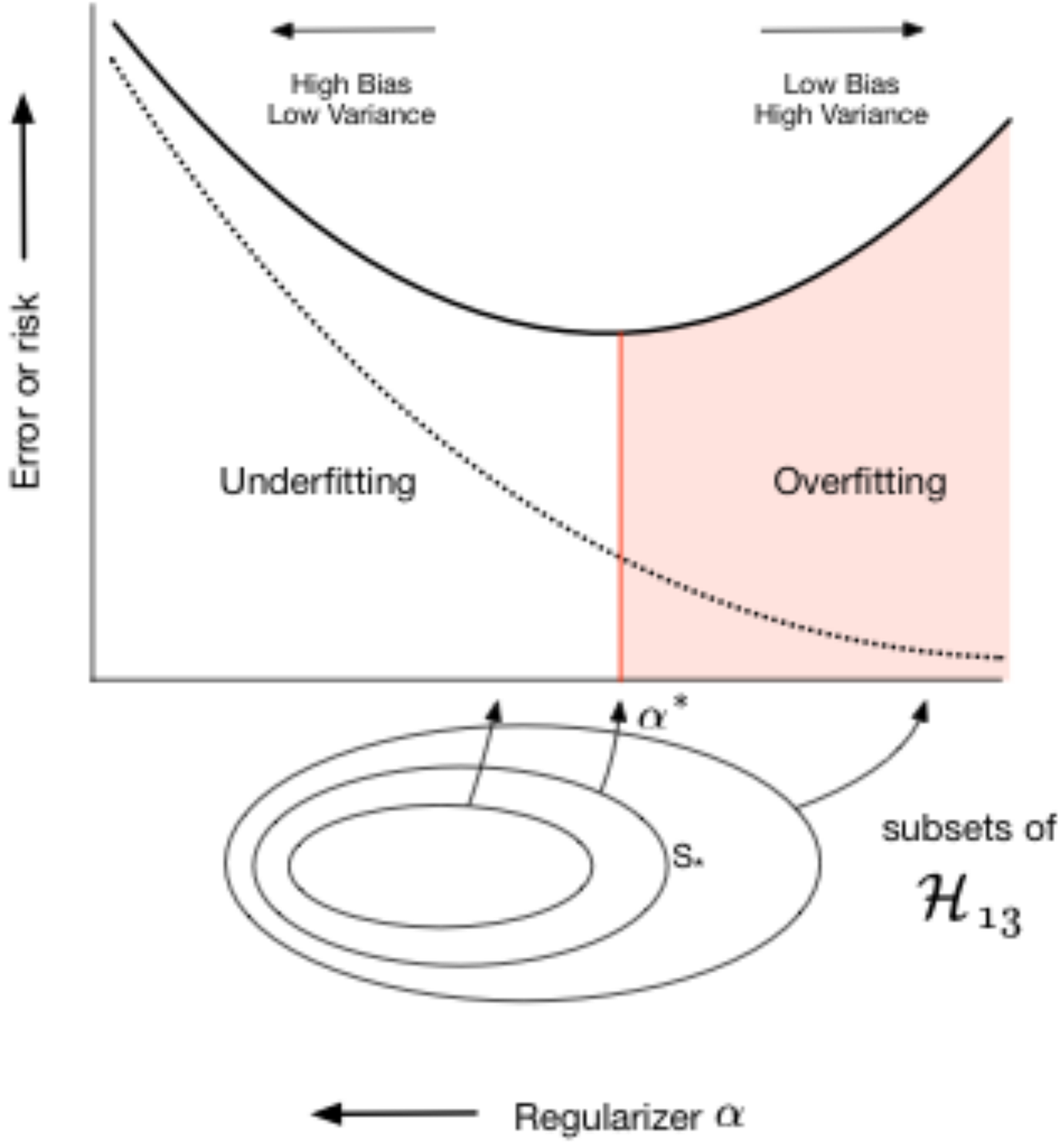
$$\mathcal{R}(h_j) = \sum_{y_i \in \mathcal{D}} (y_i - h_j(x_i))^2 + \lambda \sum_{i=0}^{j} \theta_i^2.$$

As we increase λ, coefficients go towards 0.

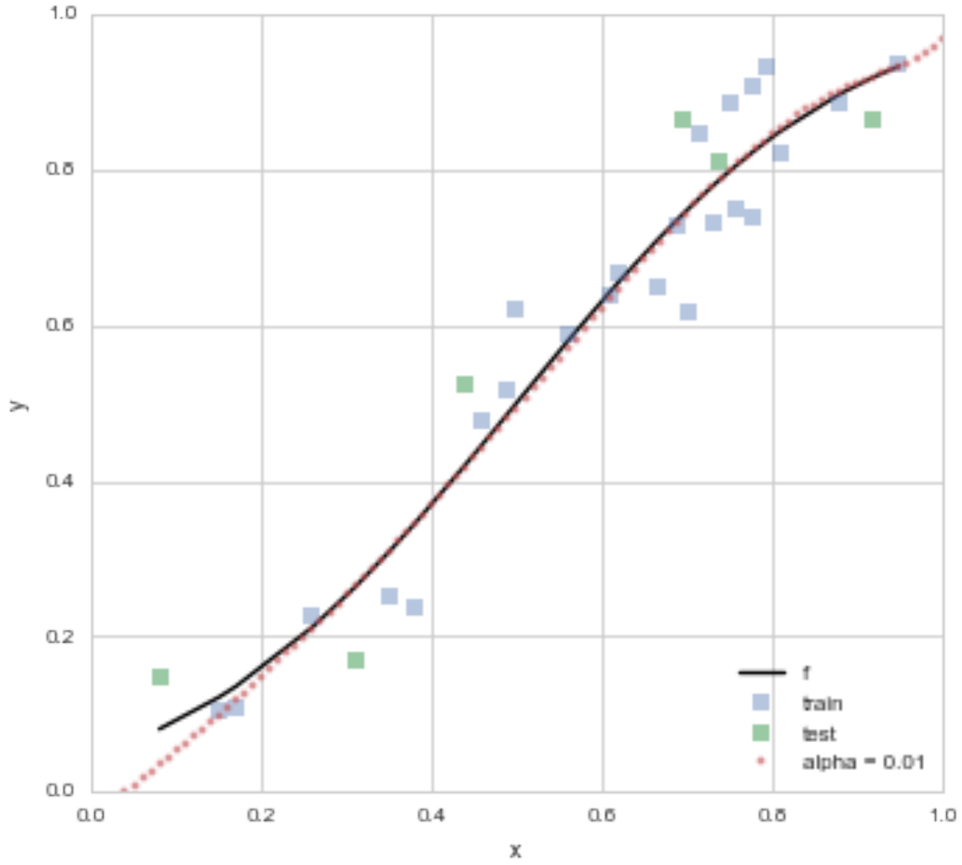
Lasso uses $\lambda \sum_{i=0}^{j} |\theta_i|$, sets coefficients to exactly 0.

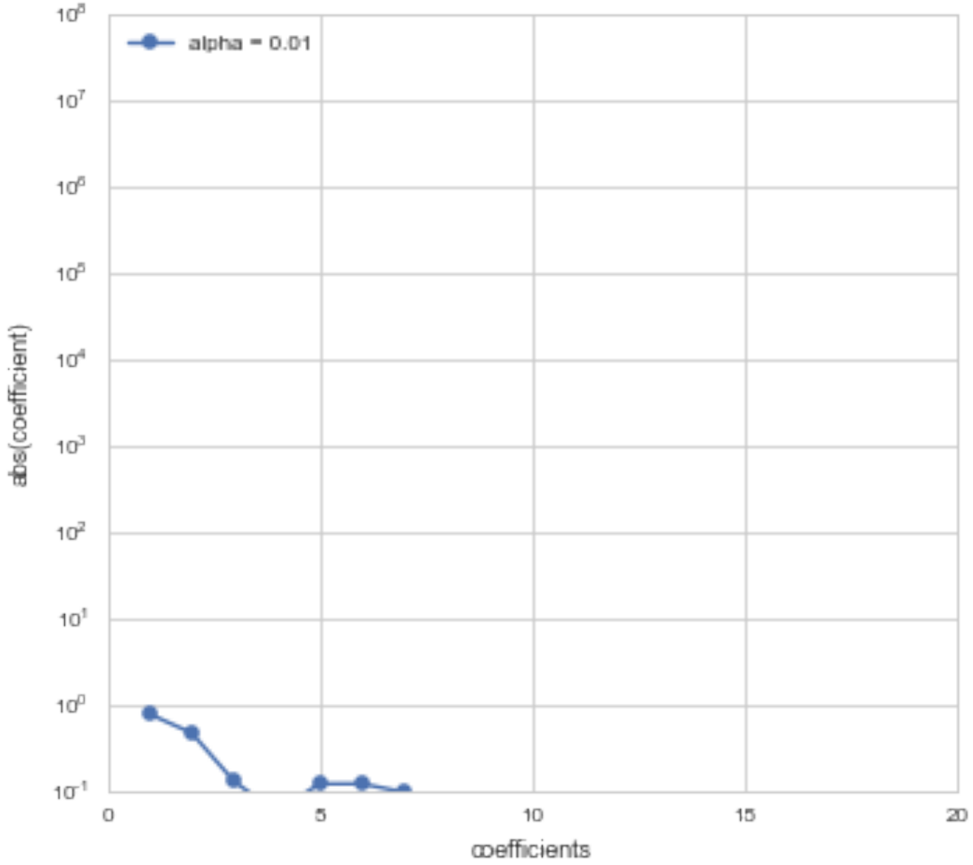
Structural Risk Minimization

- Regularization is a subsetting now,
- of a complex hypothesis set.
- If you subset too much, you underfit
- but if you do not do it enough, you overfit

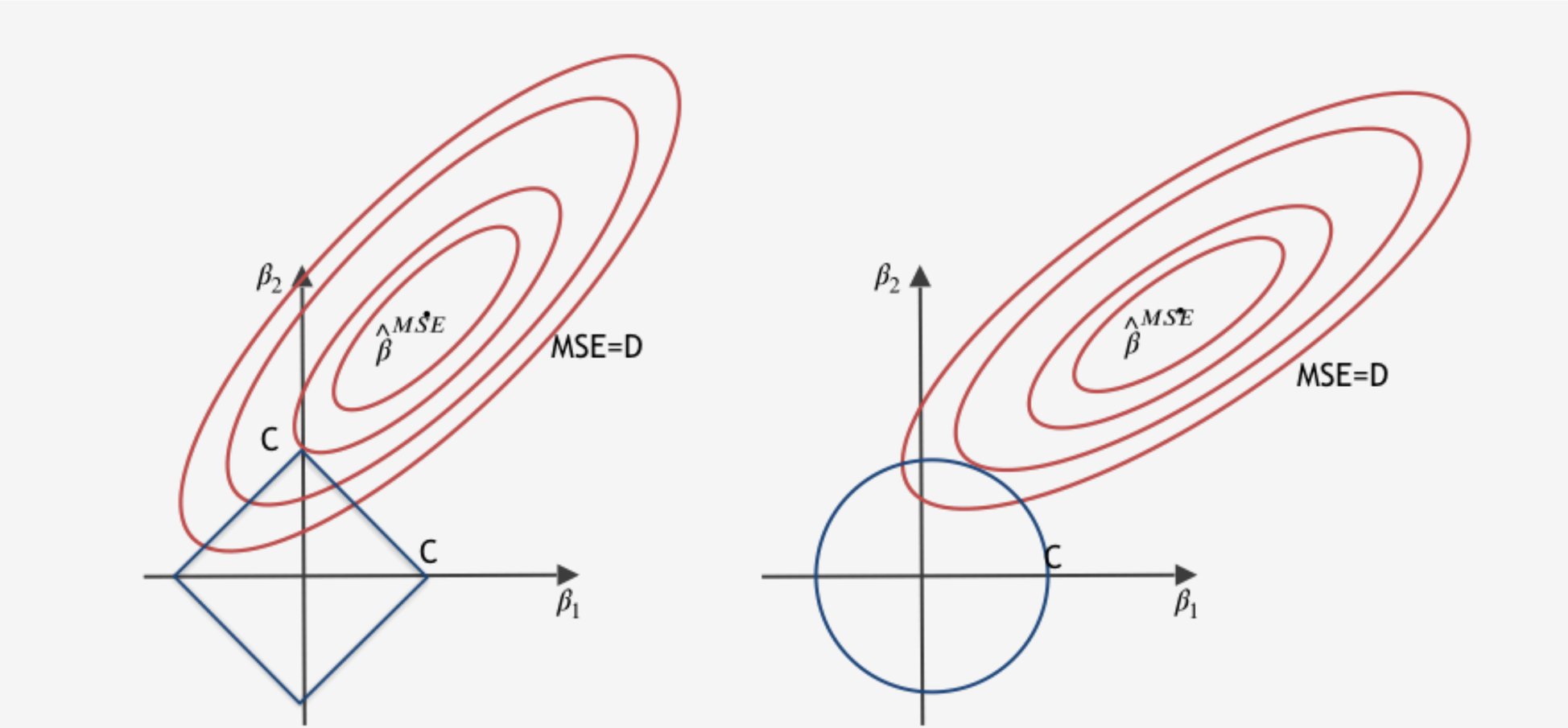


Regularization with Cross-Validation





Lasso vs Ridge Geometry



3. Lots of features

AKA: features are not just polynomial powers

Multiple Regression

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_y \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,J} \\ 1 & x_{2,1} & \dots & x_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & \dots & x_{n,J} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_J \end{pmatrix}$$

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

$$\mathrm{MSE}(\beta) = \tfrac{1}{n}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|^2$$

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y} = \operatorname*{argmin}_{\beta} \mathrm{MSE}(\boldsymbol{\beta})$$

Colinearity and co-efficients

Three individual models

	A	B	C	D	E	F
	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
	6.679	0.478	13.957	2.804e-31	5.735	7.622
	0.048	0.0027	17.303	1.802e-41	0.042	0.053

RADIO

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
	9.567	0.553	17.279	2.133e-41	8.475	10.659
	0.195	0.020	9.429	1.134e-17	0.154	0.236

NEWS

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
	11.55	0.576	20.036	1.628e-49	10.414	12.688
	0.074	0.014	5.134	6.734e-07	0.0456	0.102

One model

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
	2.602	0.332	7.820	3.176e-13	1.945	3.258
	0.046	0.0015	29.887	6.314e-75	0.043	0.049
	0.175	0.0094	18.576	4.297e-45	0.156	0.194
	0.013	0.028	2.338	0.0203	0.008	0.035

Boolean and Categorical Variables: One Hot Encoding

Income	Limit	Rating	Cards	Age	Education	Gender	Student	Married	Ethnicity	Balance
14.890	3606	283	2	34	11	Male	No	Yes	Caucasian	333
106.02	6645	483	3	82	15	Female	Yes	Yes	Asian	903
104.59	7075	514	4	71	11	Male	No	No	Asian	580
148.92	9504	681	3	36	11	Female	No	No	Hispanic	964
55.882	4897	357	2	68	16	Male	No	Yes	Caucasian	331

If the predictor takes only two values, then we create an indicator or dummy variable that takes on two possible numerical values. If more than 2 values, then need N-1 columns:

Ethnicity = {Caucasian, Asian, Hispanic} $\rightarrow$ Ethnicity = {Caucasian *or* not, Asian *or* not}

Regression with categorical variables

$$x_{i,1} = \begin{cases} 1 & \text{if } i \text{ th person is Asian} \\ 0 & \text{if } i \text{ th person is not Asian} \end{cases}$$

$$x_{i,2} = \begin{cases} 1 & \text{if } i \text{ th person is Caucasian} \\ 0 & \text{if } i \text{ th person is not Caucasian} \end{cases}$$

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i \text{ th person is Asian} \\ \beta_0 + \beta_2 + \epsilon_i & \text{if } i \text{ th person is Caucasian} \\ \beta_0 + \epsilon_i & \text{if } i \text{ th person is Hispanic} \end{cases}$$

What do we mean by linear?

We presented polynomial regression as if it was not linear regression. But it is.

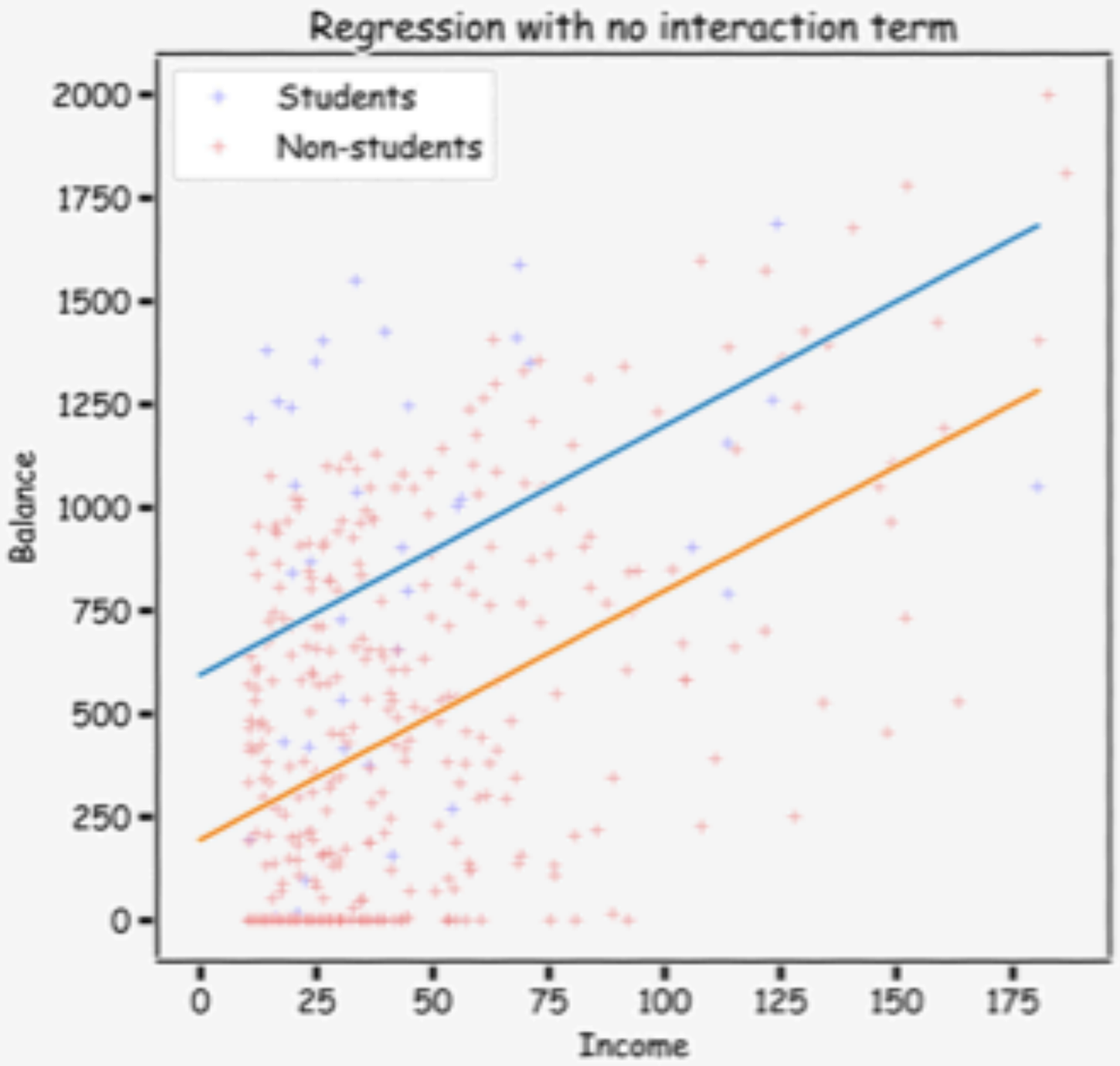
Linearity refers to the coefficients, bot the features.

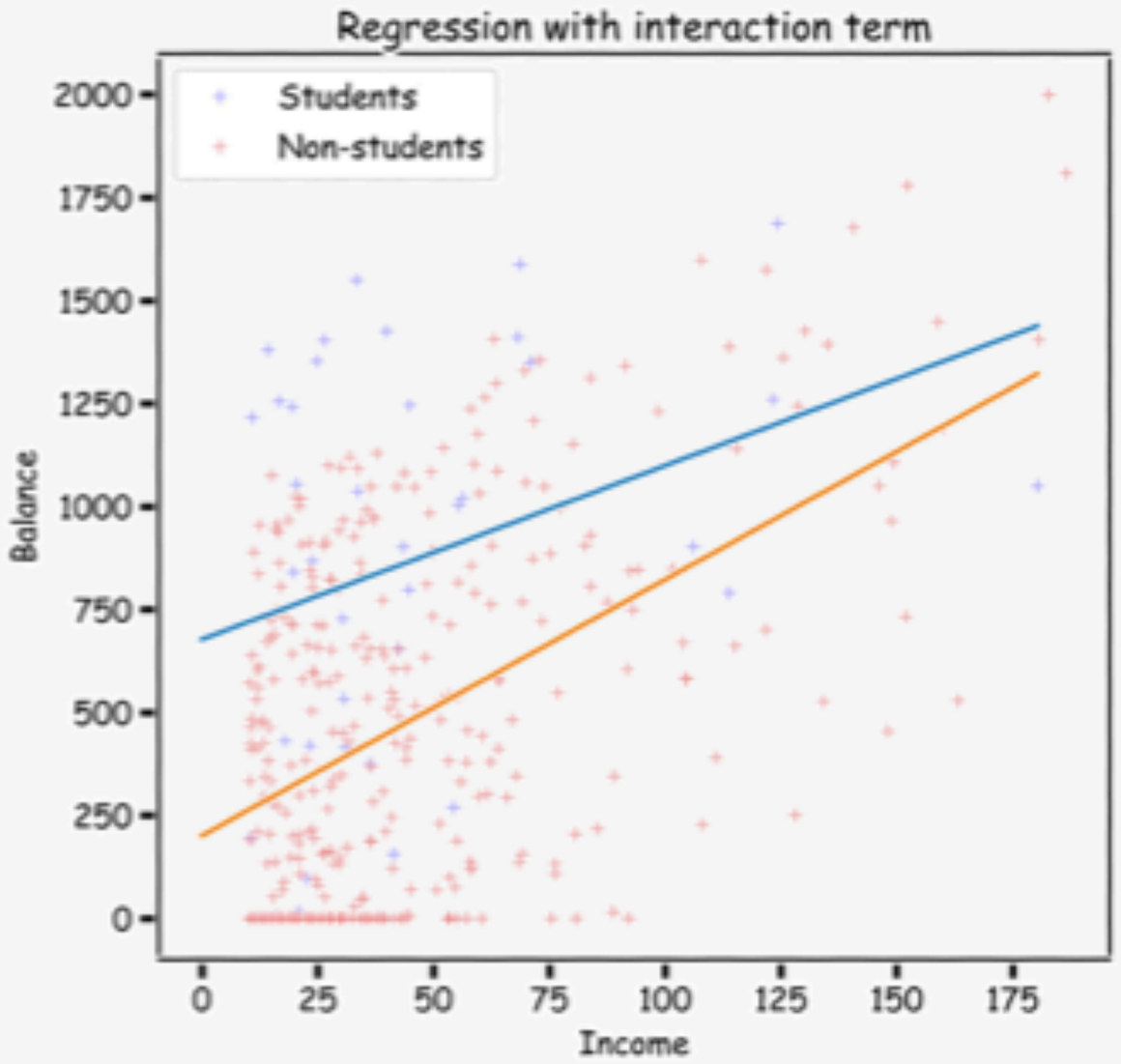
Here is another example: interaction terms with a categorical variable:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$$

Here we interact X_1 and X_2. What does this mean?


Regression with no interaction term
Students
Non-students
2000
1750
1500
1250
1000
750
500
250
0
Balance
0
25
50
75
100
125
150
175
Income

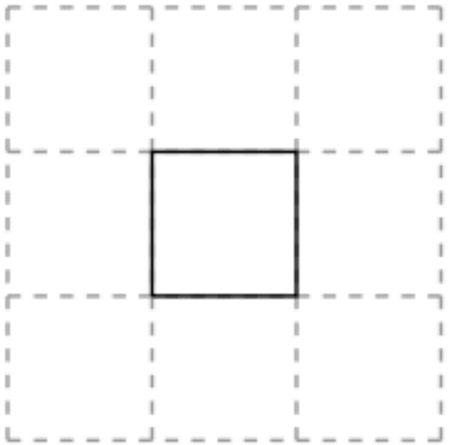

Regression with interaction term
Students
Non-students
2000
1750
1500
1250
1000
750
500
250
0
Balance
0
25
50
75
100
125
150
175
Income

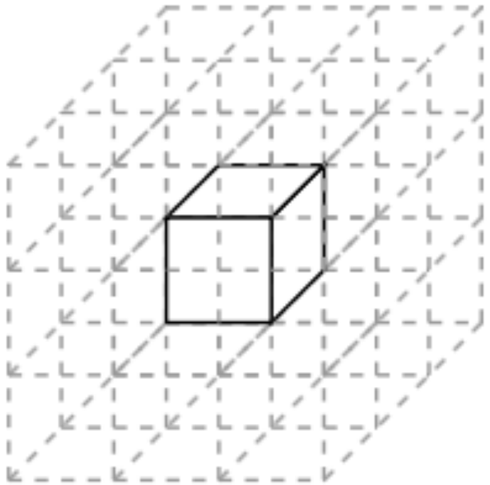
As you can see, the number of features can balloon. In many modern problems: startup with few customers but lots of data on them, there are already more predictors than members in your sample.

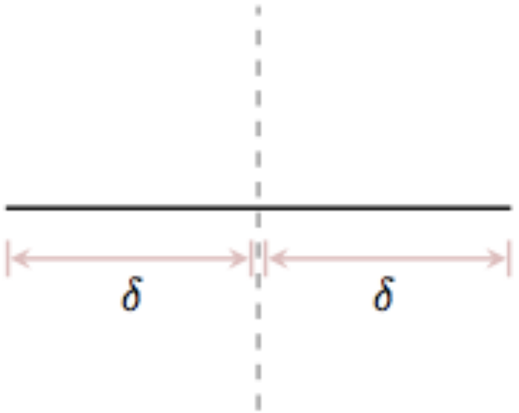
We then get the :

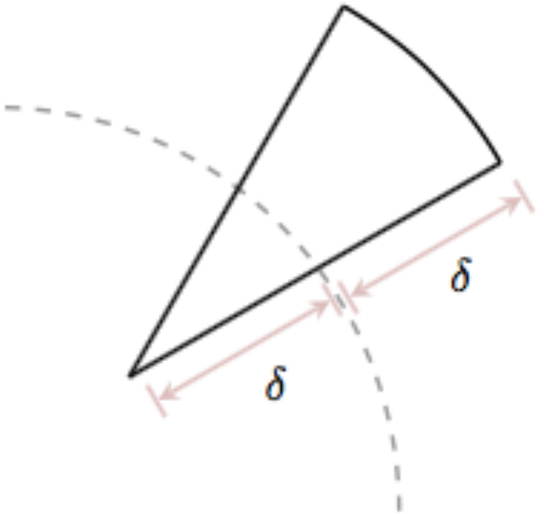
Curse of Dimensionality

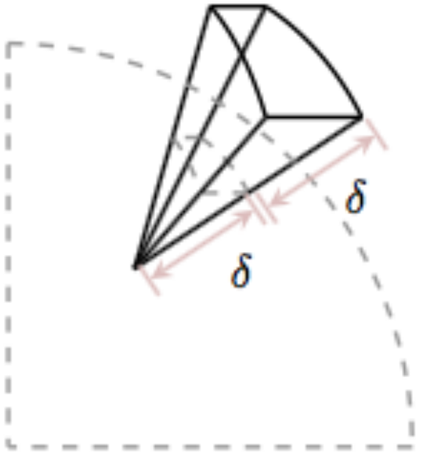
- data is sparser in higher dimensions
- volume moves to the outside



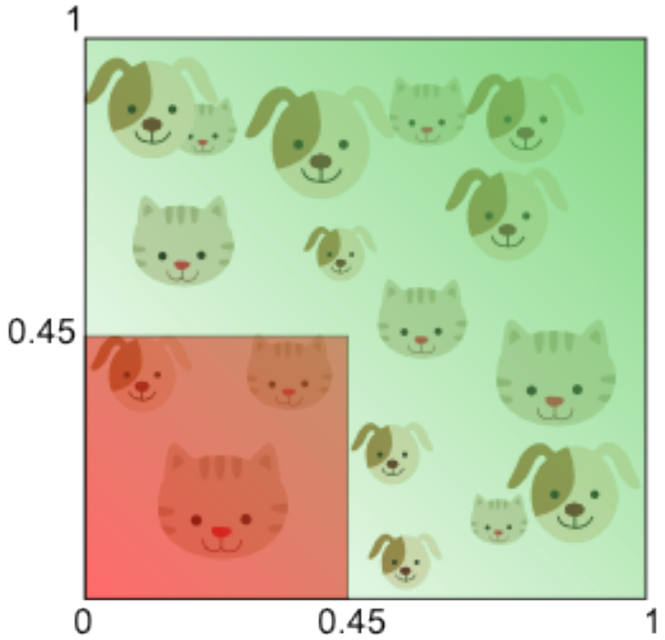


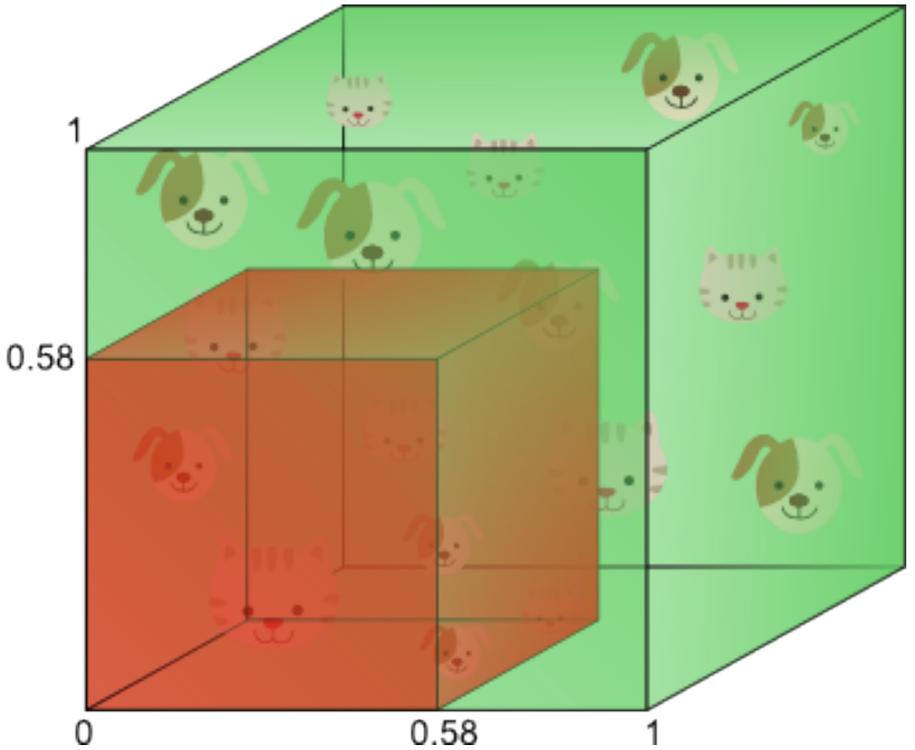

δ
δ


δ
δ


δ
δ

to cover same fractional volume, you need to go bigger on length in higher dims





Overfitting and the curse

- remember dimensionality in our problems refers to the number of features we have
- each feature (or feature combination which we shall just call a new feature) is a dimension
- thus each member of our sample is a point in this feature space
- notions of distance and volume become hard in this high-dimensional space
- indeed its easier to find "simple models" in this high dimensional space