Lecture 6

Risk and Information Theory

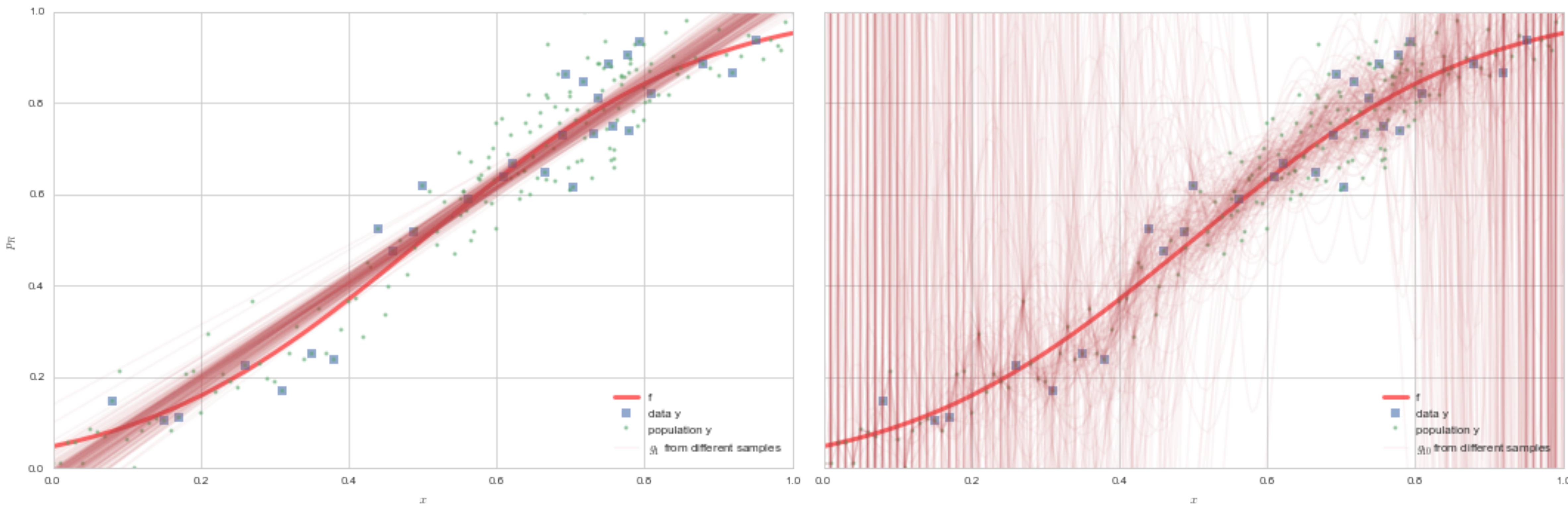
Last Time:

- Normal MLE and Regression
- Test Sets
- Validation and X-validation
- Regularization

Today

- Risk and Bayes Risk
- The KL Divergence and Deviance
- In-sample penalties: the AIC
- Entropy
- Maximum Likelihood and Entropy

UNDERFITTING (Bias)
vs OVERFITTING (Variance)



Sources of Variability

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

	generate : x^2 fit : x^2	generate : x^2 fit : x
	fixed x	sample x
no ϵ	no $p(y\|x)$ no $p(y\|x)$	no $p(y\|x)$ $\boxed{p(y\|x) \text{ only from sampling}}$ ← deterministic noise contrib to variance
ϵ	$p(y\|x)$ only from ϵ $p(y\|x)$ only from ϵ	$p(y\|x)$ from both. $p(y\|x)$ from both.

Risk for a given h

Define:

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

$$= \int dydxp(y \mid x)p(x)(h(x) - y)^2 = E_X E_{Y|X}[(h - y)^2].$$

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

(we assume 0 mean finite-variance noise ϵ)

- Varying training sets make empirical $R_{out}(h)$ a stochastic quantity, varying from one training set to another.

- This can be written as:

$$R_{out}(\hat{h}_n) = E_{p(x,y)}[(h(x) - y)^2 \mid \hat{h}_n]$$
$$= \int dx p(x,y)(\hat{h}_n(x) - y)^2.$$

- Average empirical risk over the training sets (a different model is fit on each set)

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

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

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

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

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

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

Empirical Risk Minimization

- LLN suggests that we can replace the risk integral by a data sum and then minimize
- Assume $(x_i, y_i) \sim P(x, y)$ (use empirical distrib)
- Fit hypothesis $h = g_{\mathcal{D}}$, where $\mathcal{D}$ is our training sample.
- $R_{out}(g_{\mathcal{D}}) = \sum_{i \in \mathcal{D}} (g_i - y_i)^2$
- minimize to get best for $g_{\mathcal{D}}$

$$R(h) = E_{XY}[L(h, y)]$$

$$\hat{R}_n = \frac{1}{N} \sum_i L(y_i, h(x_i))$$

For each h LLN implies convergence from empirical to actual.

Now, $R^* = \inf_{all h} R(h)$ becomes infimum over empirical risks. But again restrict to $\mathcal{H}$ otherwise overfitting!

- Varying training sets make empirical $R_{out}(h)$ a stochastic quantity, varying from one training set to another.
- Thus average empirical risk over the training sets (a different model is fit on each set)
- **Goal of Learning**: Build a function whose risk is closest to Bayes Risk

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

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

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

where $y = f(x) + \epsilon$ is the true generating process and ϵ has 0 mean and finite variance σ^2.

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

This is the bias variance decomposition for regression.

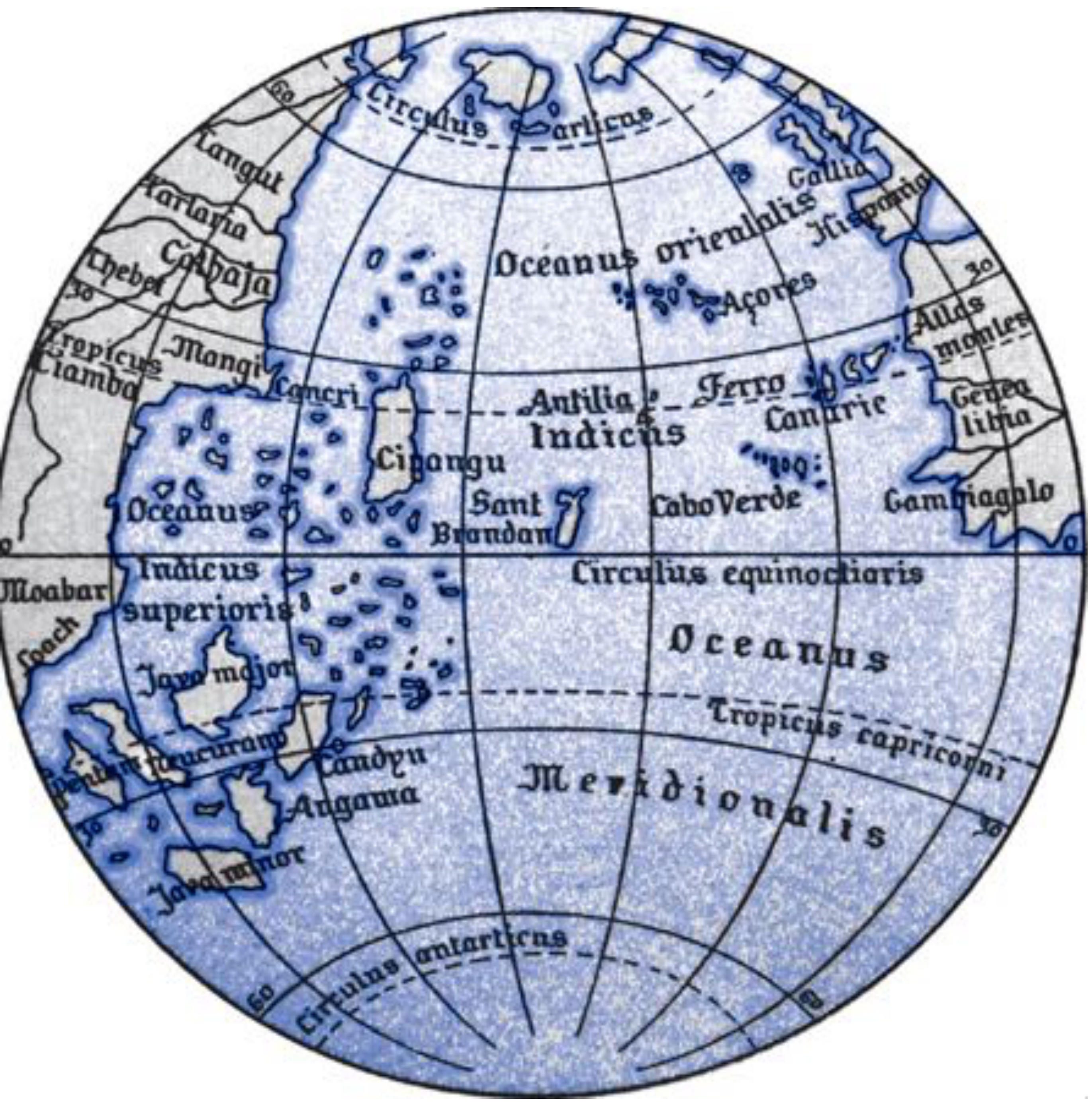
Or, written as $\langle R \rangle - R^*$, this is

`variance` + `bias`^{2}, or

`estimation-error` + `approximation-error`

$$R(g) - \inf_{g \in \mathcal{H}} R(g) + \inf_{g \in \mathcal{H}} R(g) - R^*$$

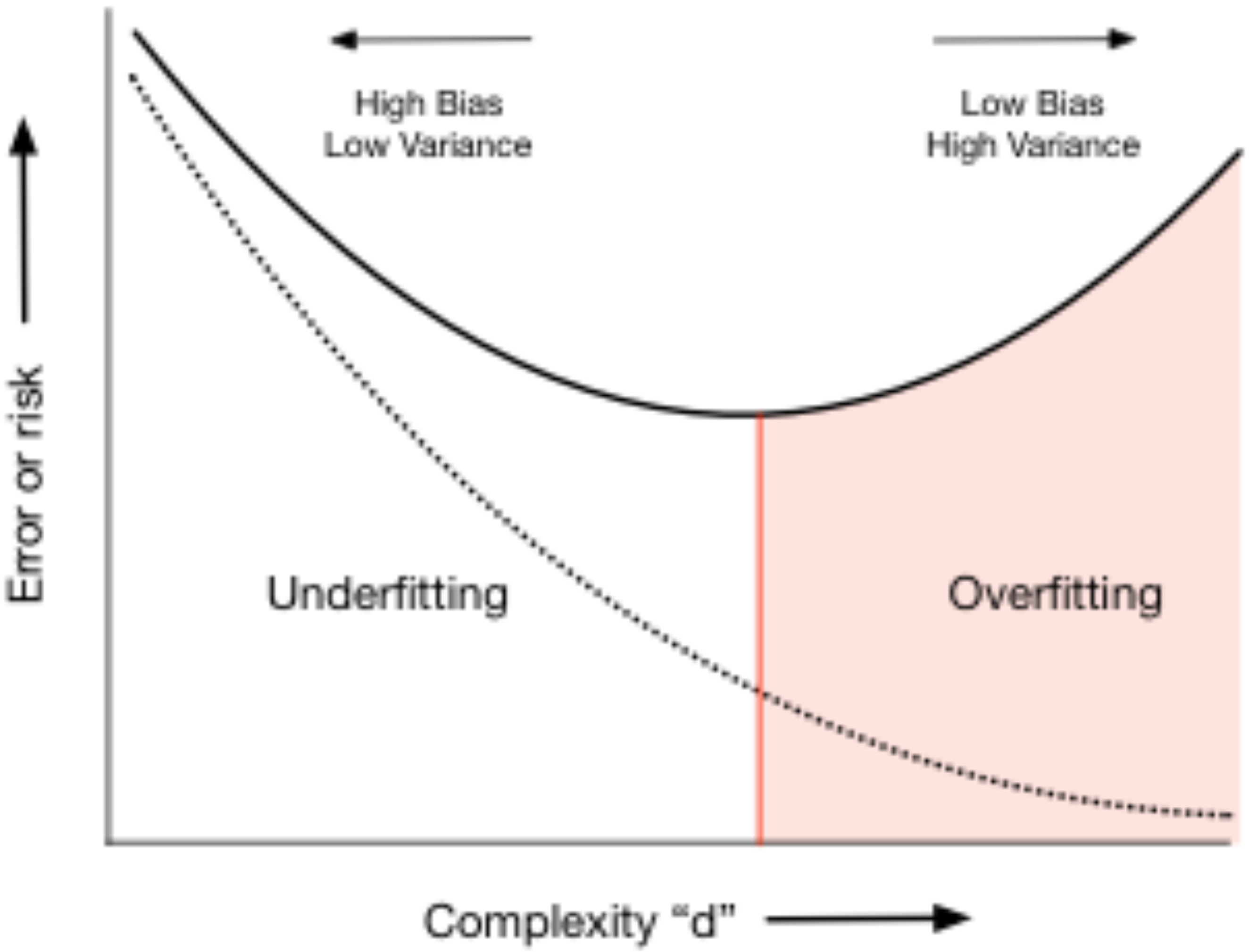
- first term is **variance**, squared error of the various fit g's from the average g, the hairiness.
- second term is **bias**, how far the average g is from the original f this data came from.
- third term is the **stochastic noise**, minimum error that this model will always have.

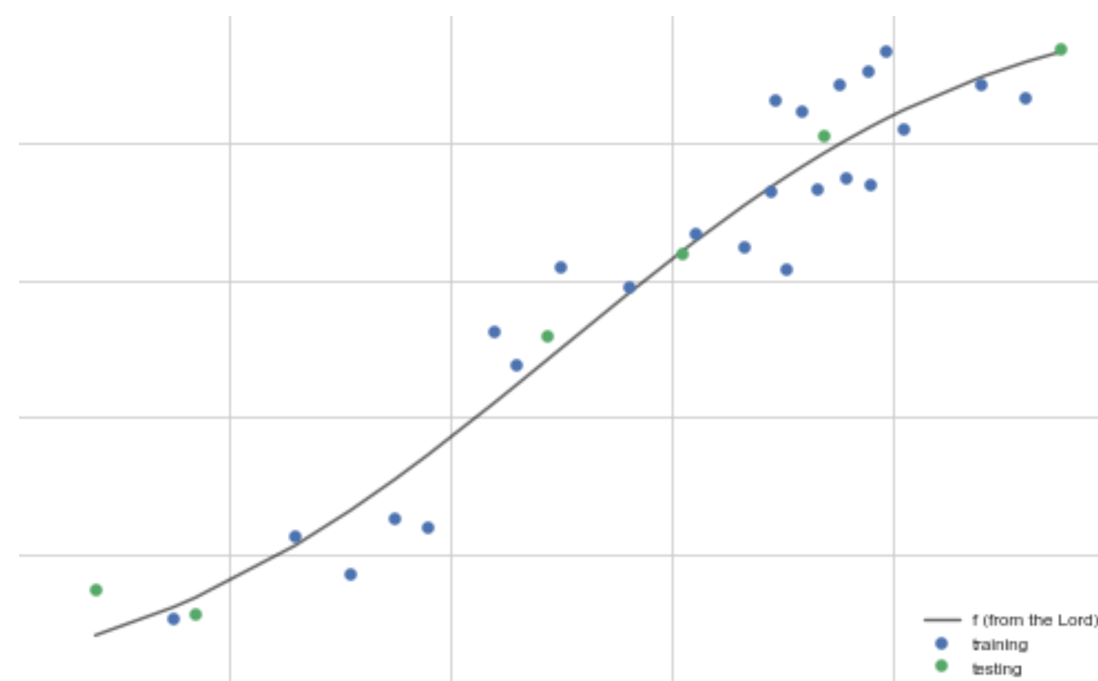


SMALL World vs BIG World

- *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".

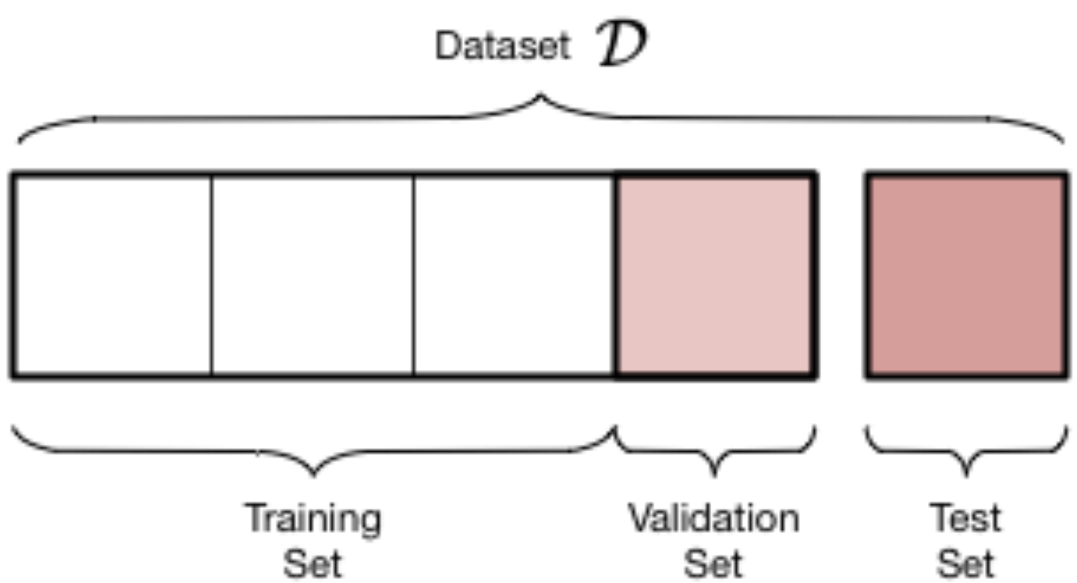
MODEL COMPARISON: A Large World approach

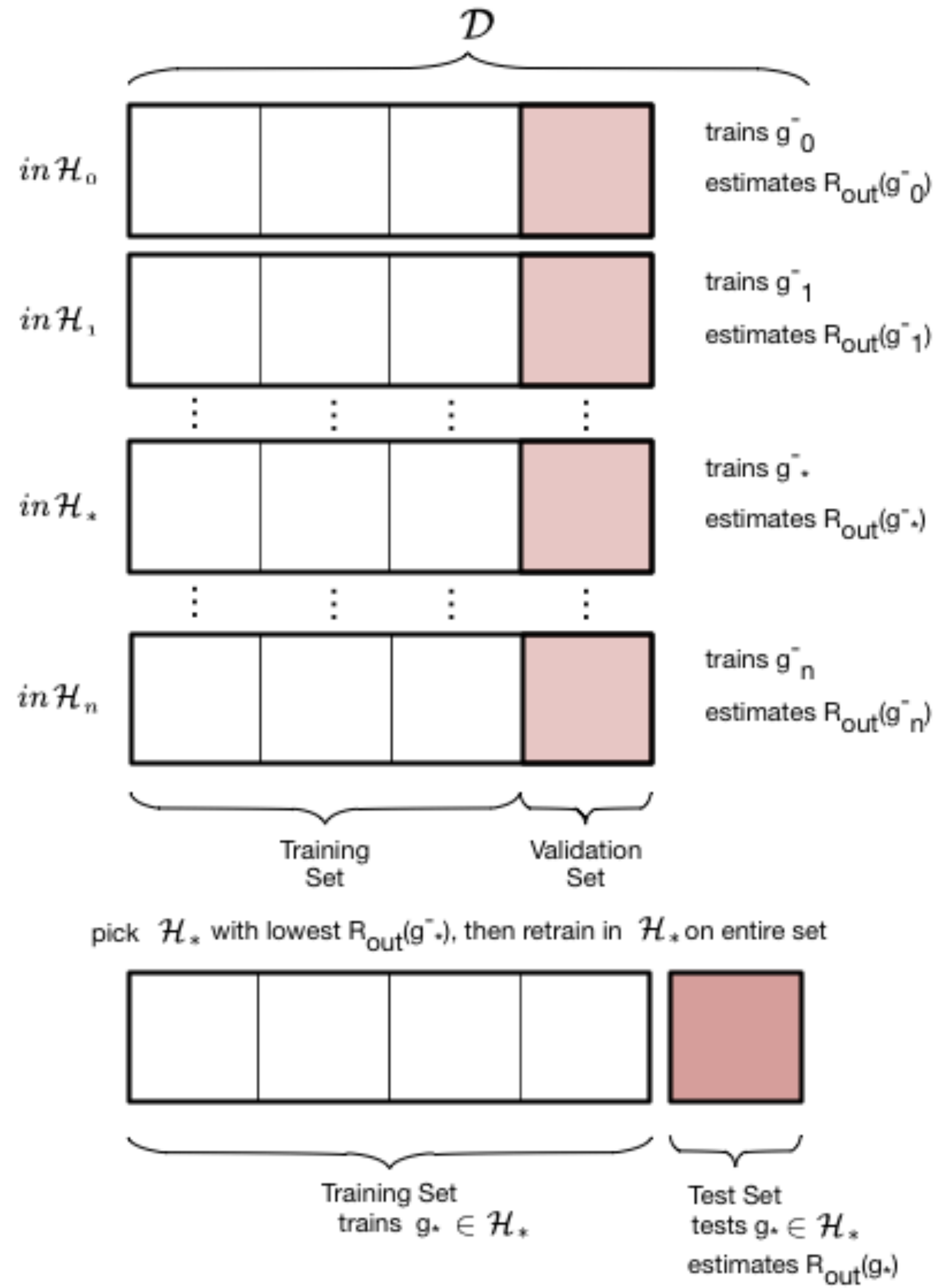




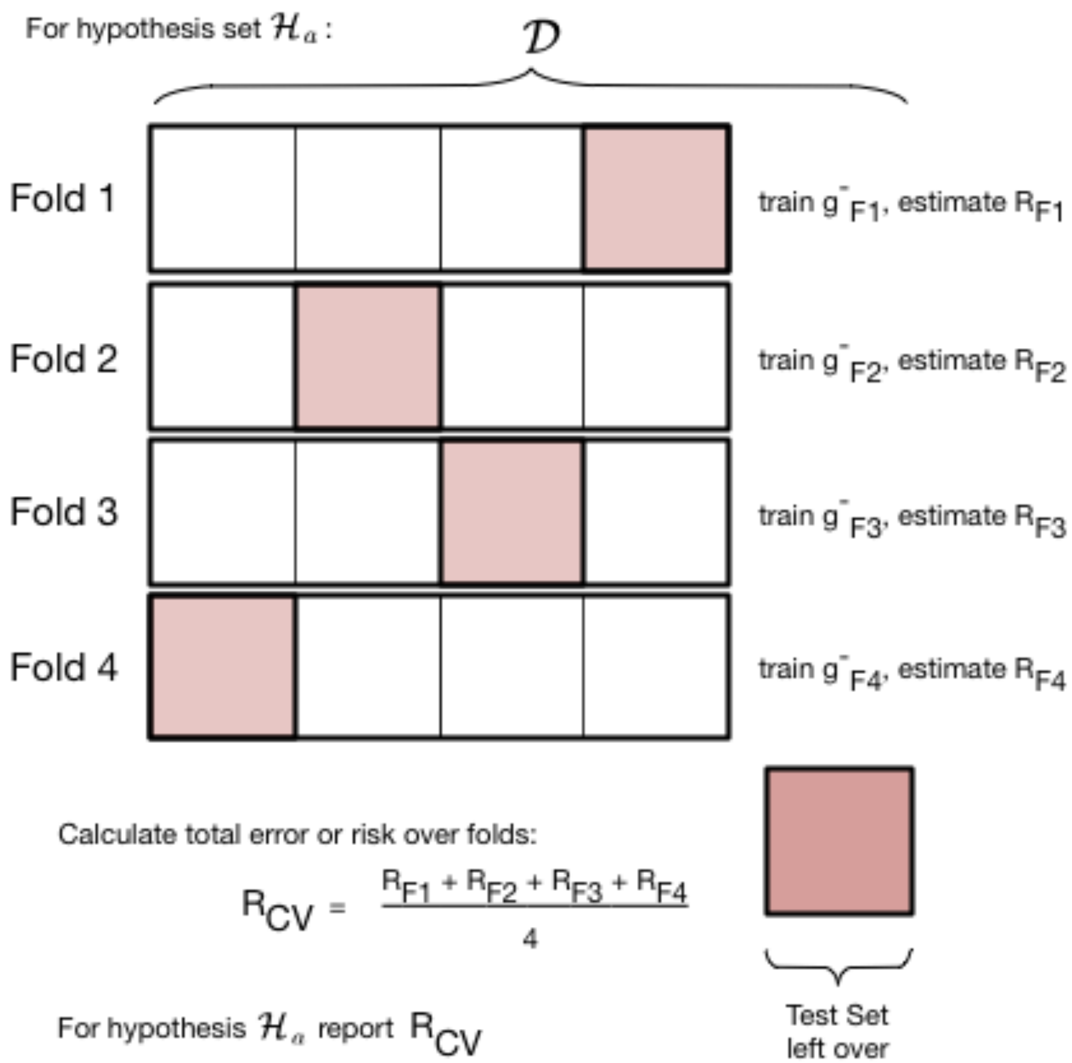
VALIDATION

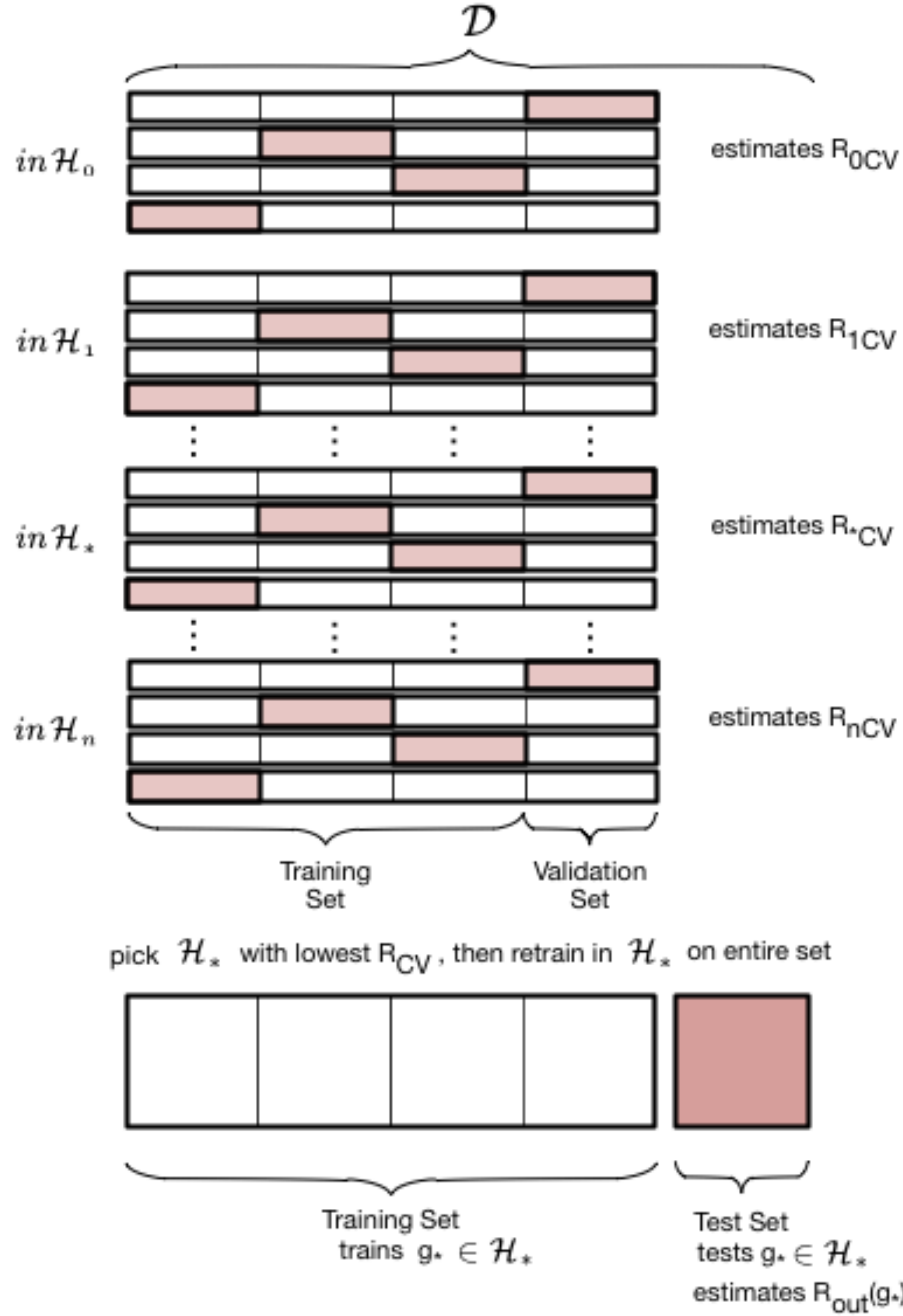
- train-test not enough as we *fit* for d on test set and contaminate it
- thus do train-validate-test





CROSS-VALIDATION





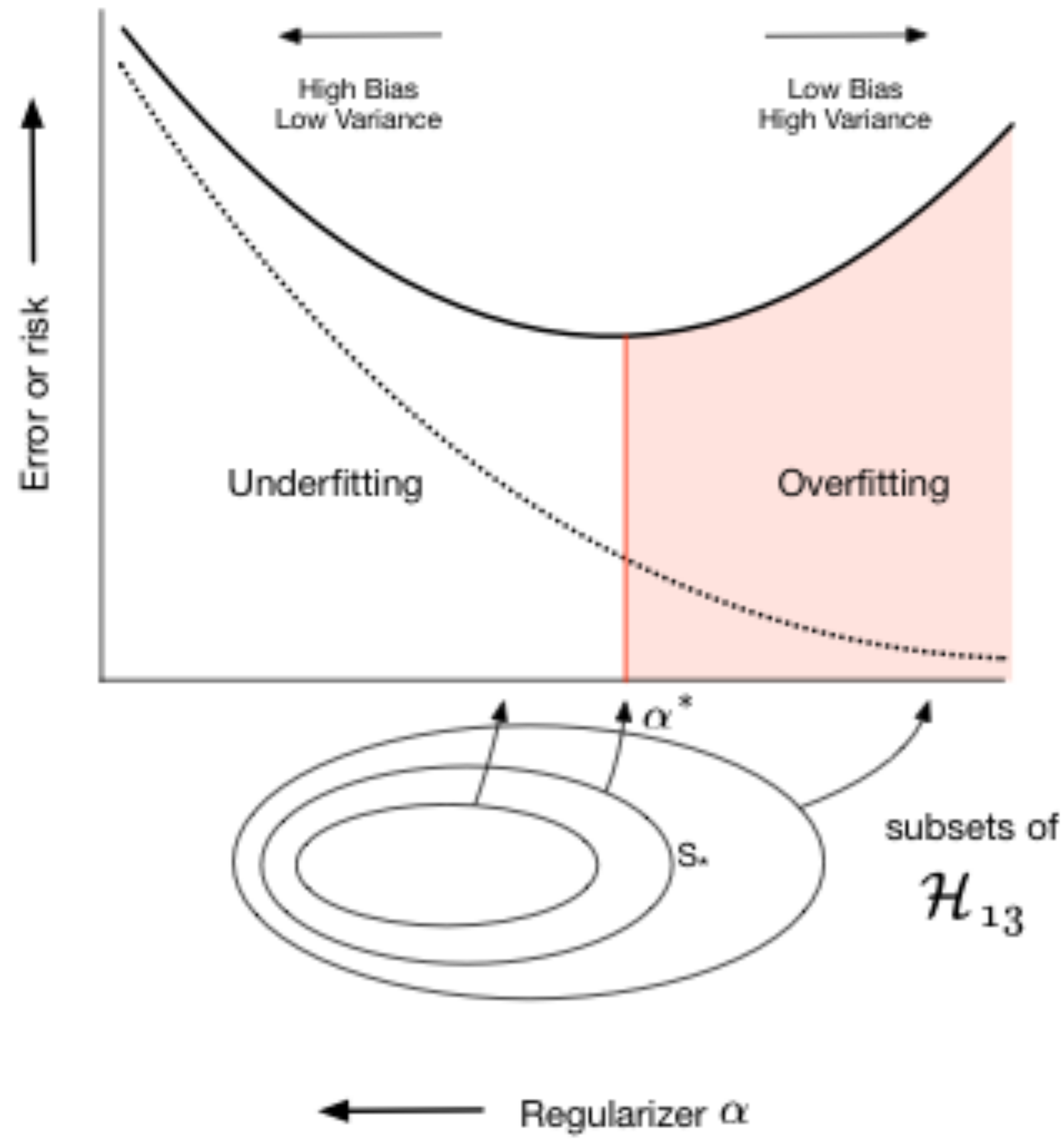
REGULARIZATION: A SMALL WORLD APPROACH

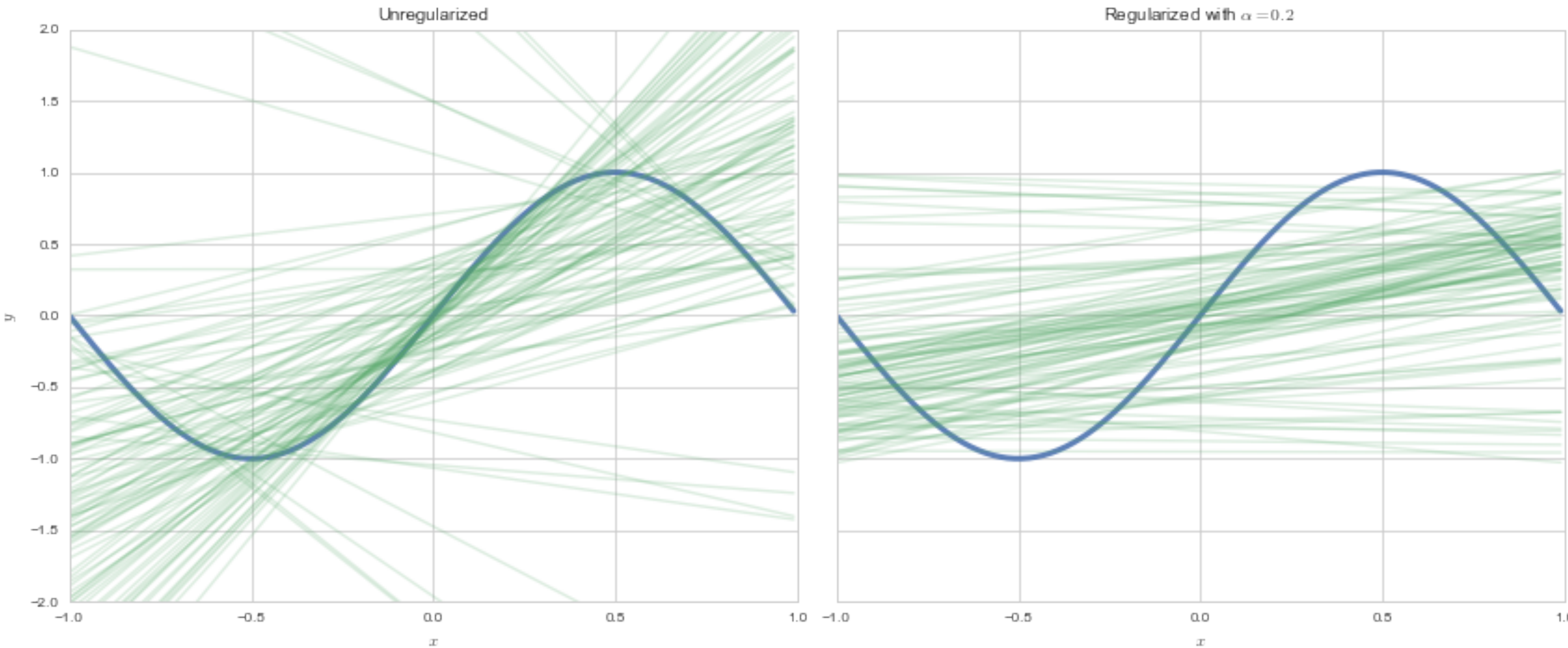
Keep higher a-priori complexity and impose a

complexity penalty

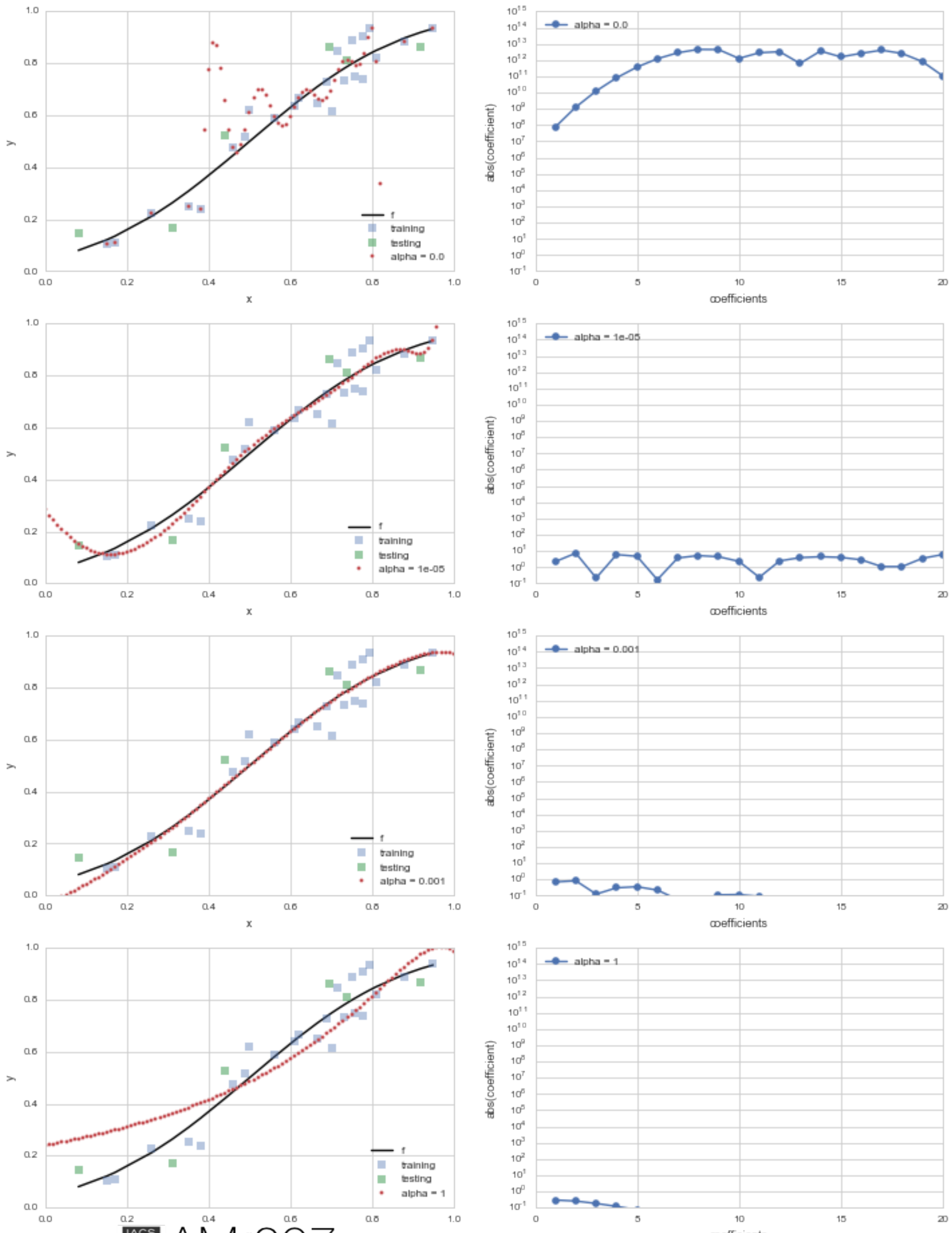
on risk instead, to choose a SUBSET of $\mathcal{H}_{big}$. We'll make the coefficients small:

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





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



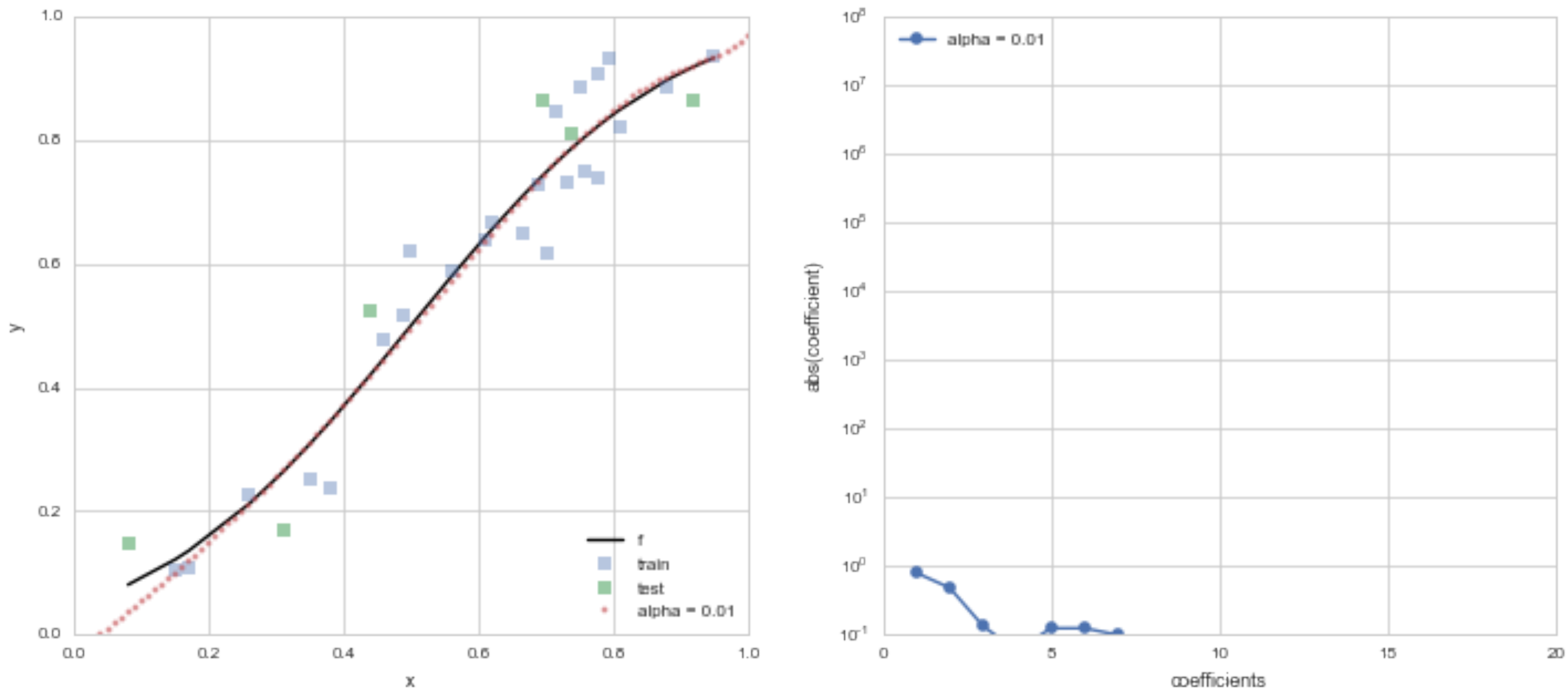
REGULARIZATION

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

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



MODEL COMPARISON: In-sample estimation

- Suppose we have a large-world subset of nested models.
- .. thus the models have the same likelihood form
- would be nice to not have to spend data on validation sets
- and exploit the notion that a negative log likelihood is a loss
- we could use strength of effects
- but not really needed for prediction

KL-Divergence

$$D_{KL}(p, q) = E_p[log(p) - log(q)] = E_p[log(p/q)]$$
$$= \sum_i p_i log(\frac{p_i}{q_i}) \ or \int dPlog(\frac{p}{q})$$

$$D_{KL}(p, p) = 0$$

KL divergence measures distance/dissimilarity of the two distributions $p(x)$ and $q(x)$.

Divergence:

The additional uncertainty induced by using probabilities from one distribution to describe another distribution

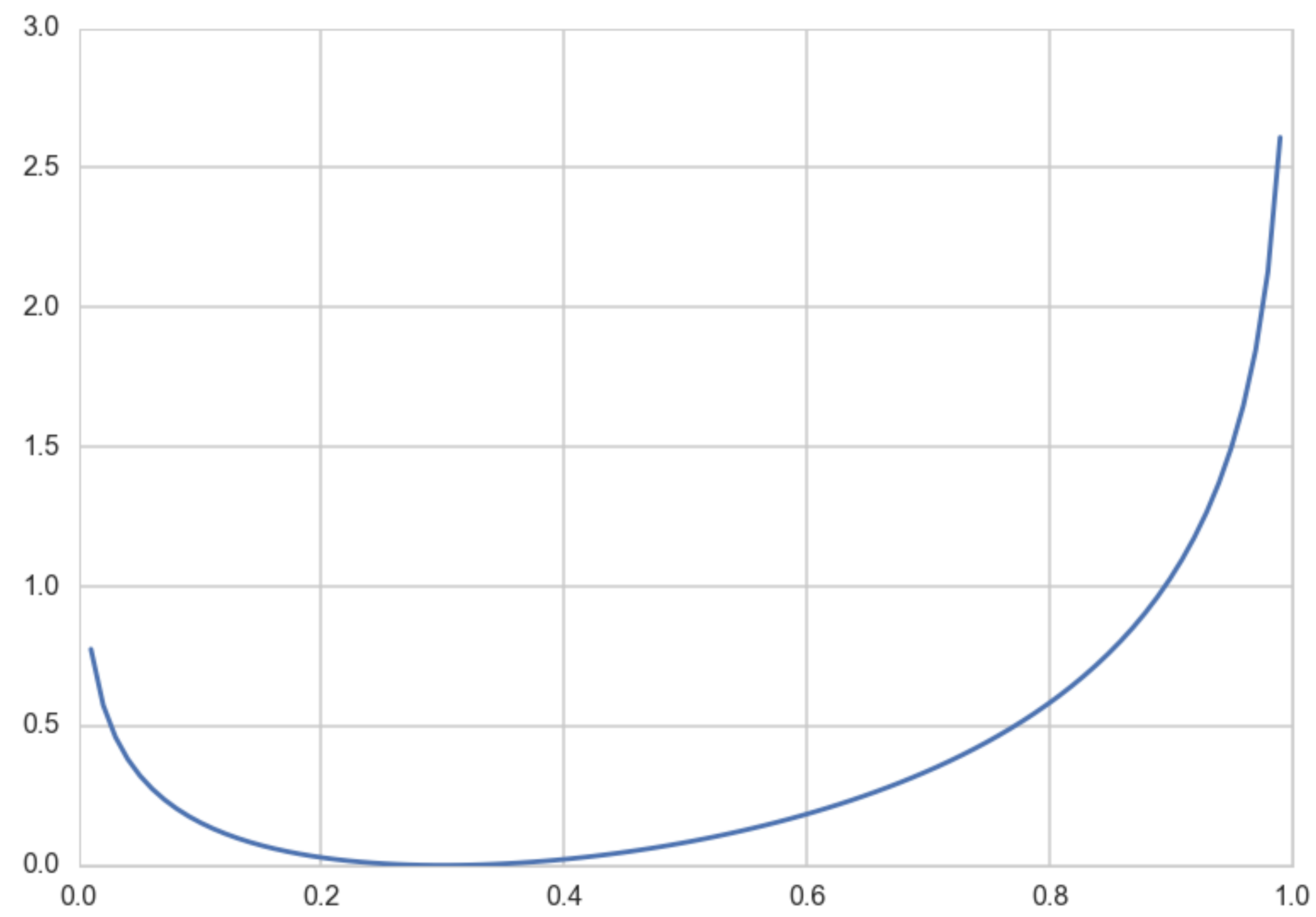
- McElreath page 179

KL example

Bernoulli Distribution p with
$p = 0.3.$

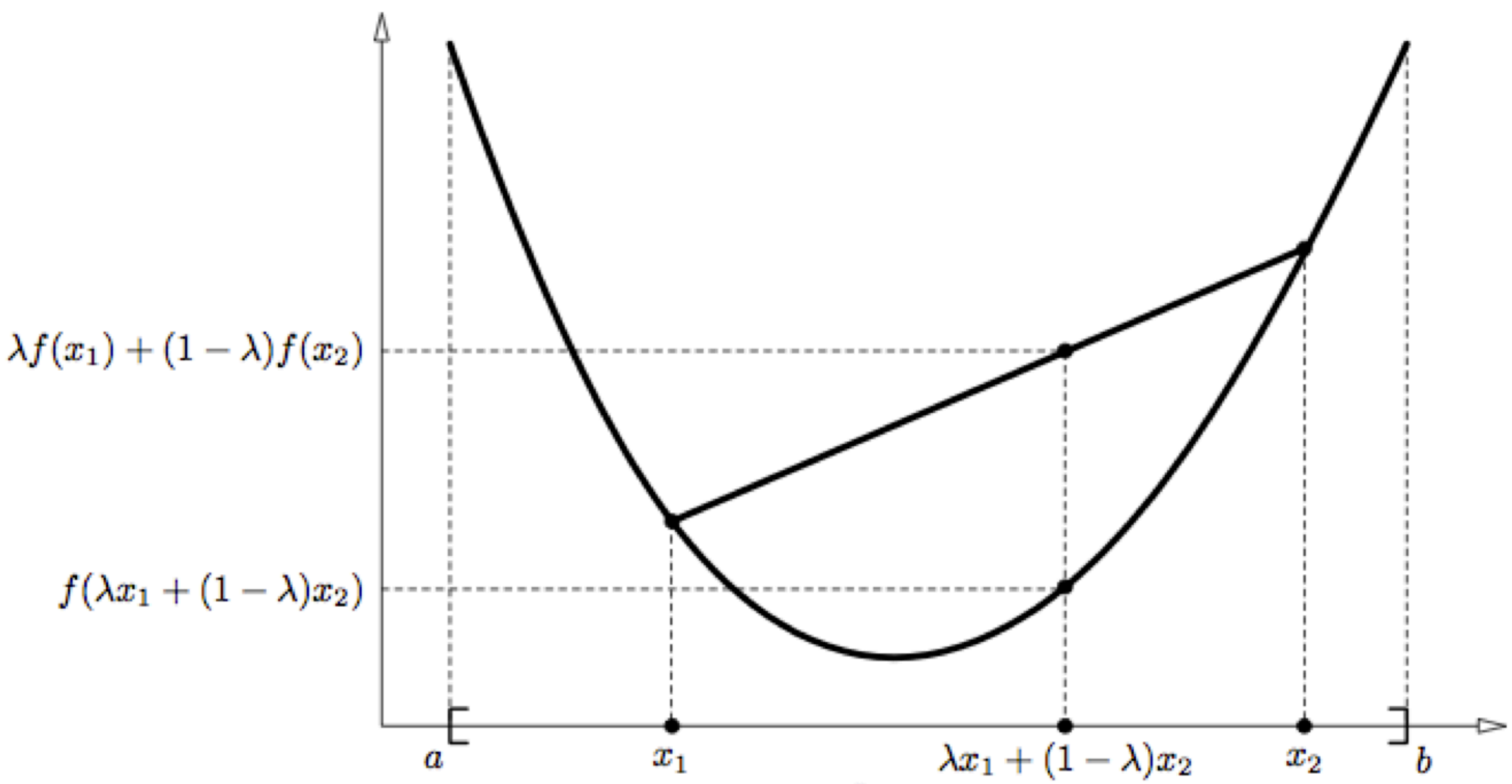
Try to approximate by q. What parameter?

```
def kld(p,q):
    return p*np.log(p/q) + (1-p)*np.log((1-p)/(1-q))
```



Jensen's Inequality for convex $f(x)$:

$$E[f(X)] \geq f(E[X])$$



KL-Divergence is always non-negative

Jensen's inequality:

$$\implies D_{KL}(p, q) \geq 0 \text{ (0 iff } q = p \, \forall x).$$

$$D_{KL}(p, q) = E_p[log(p/q)] = E_p[-log(q/p)] \geq -\log(E_p[q/p]) =$$
$$-\log(\int dQ) = 0$$

MARS ATTACKS (Topps, 1962; Burton 1996)

$$Earth : q = \{0.7, 0.3\}, Mars : p = \{0.01, 0.99\}.$$

Earth to predict Mars, less surprise on landing: $D_{KL}(p, q) = 1.14, D_{KL}(q, p) = 2.62$.

PROBLEM: we dont know distribution p. If we did, why do inference?

SOLUTION: Use the empirical distribution

That is, approximate population expectations by sample averages.

$$\implies D_{KL}(p, q) = E_p[log(p/q)] = \frac{1}{N}\sum_i log(p_i/q_i)$$

Maximum Likelihood justification

$$D_{KL}(p,q) = E_p[log(p/q)] = \frac{1}{N}\sum_i (log(p_i) - log(q_i))$$

Minimizing KL-divergence $\implies$ maximizing $\sum_i log(q_i)$

Which is exactly the log likelihood! MLE!

Model Comparison: Likelihood Ratio

$$D_{KL}(p,q) - D_{KL}(p,r) = E_p[log(r) - log(q)] = E_p[log(\frac{r}{q})]$$

In the sample approximation we have:

$$D_{KL}(p,q) - D_{KL}(p,r) = \frac{1}{N}\sum_i log(\frac{r_i}{q_i}) = \frac{1}{N} log(\frac{\prod_i r_i}{\prod_i q_i}) = \frac{1}{N} log(\frac{\mathcal{L}_r}{\mathcal{L}_q})$$

MODEL COMPARISON: Deviance

You only need the sample averages of the logarithm of r and q:

$$D_{KL}(p,q) - D_{KL}(p,r) = \langle log(r) \rangle - \langle log(q) \rangle$$

Define the deviance: $D(q) = -2 \sum_i log(q_i)$, a **LOSS** ...

$$D_{KL}(p,q) - D_{KL}(p,r) = \frac{2}{N}(D(q) - D(r))$$

Example

Generate data from:

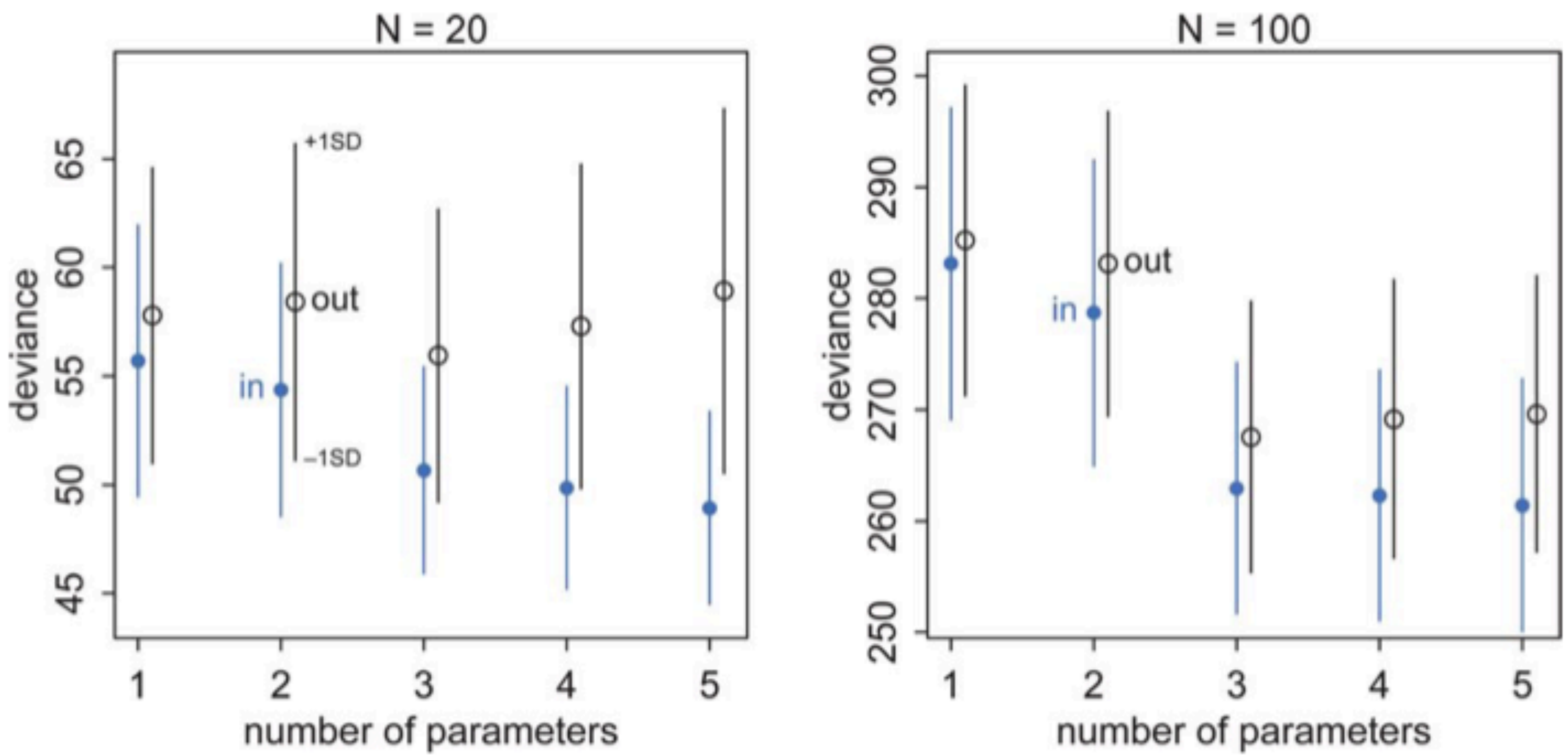
$$\mu_i = 0.15 x_{1,i} - 0.4 x_{2,i}, \;\; y \sim N(\mu, 1)$$

2 parameter model.

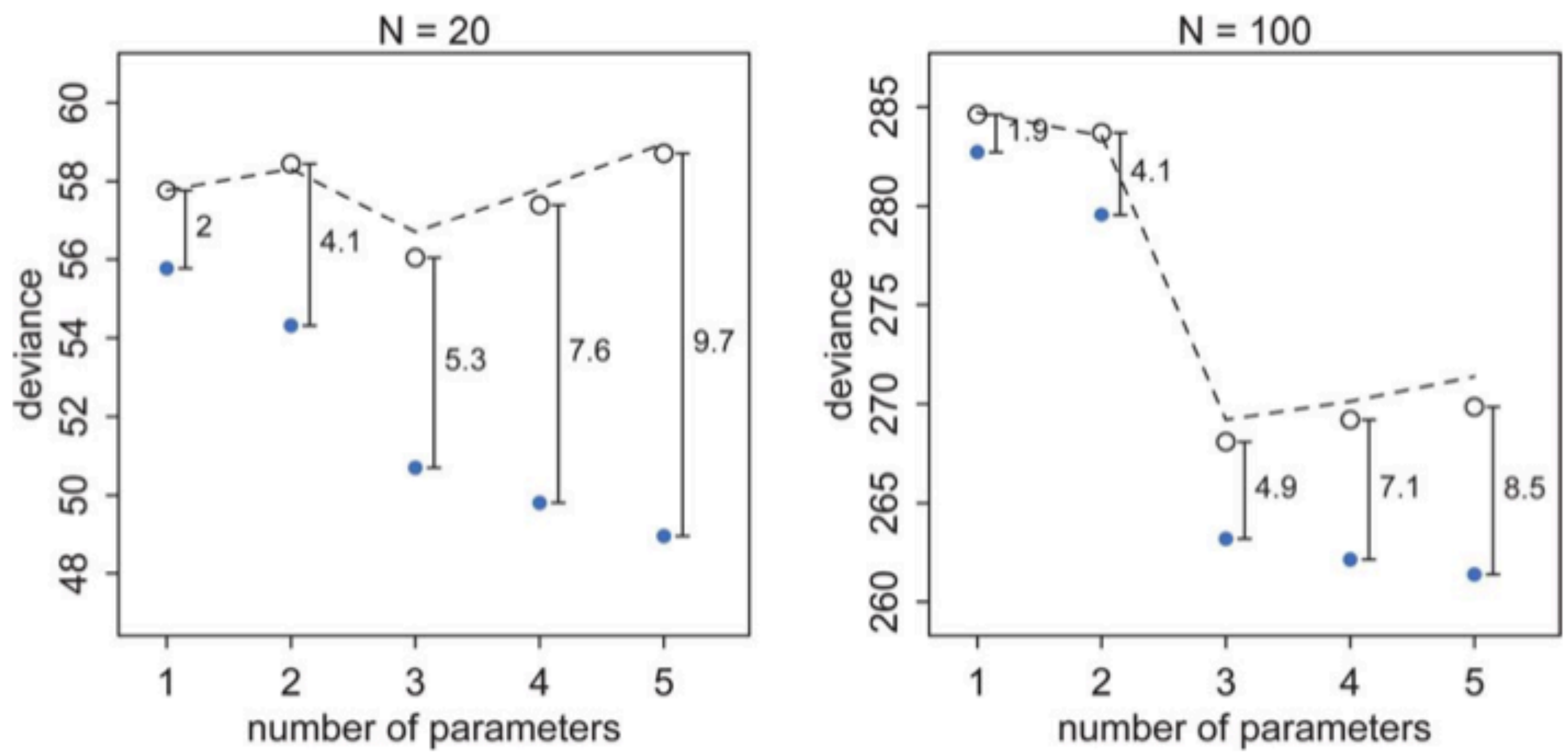
Generate 10,000 realizations, for 1-5 parameters, 20 data points and 100 data points.

Split into train and test, and do OLS.

Train and Test Deviances



Train and Test Deviances



The test set deviances are $2 * p$ above the training set ones.

Akake **Information Criterion**:

AIC **estimates out-of-sample deviance**

$$AIC = D_{train} + 2p$$

- Assumption: likelihood is approximately multivariate gaussian.
- penalized log-likelihood or risk if we choose to identify our distribution with the likelihood: REGULARIZATION

AIC for Linear Regression

$$AIC = D_{train} + 2p \text{ where}$$

$$D(q) = -2\sum_i log(q_i) = -2\ell$$

$$\sigma^2_{MLE} = \frac{1}{N} SSE$$

$$AIC = -2(-\frac{N}{2}(log(2\pi) + log(\sigma^2)) - 2(-\frac{1}{2\sigma^2_{MLE}} \times SSE) + 2p$$

$$AIC = Nlog(SSE/N) + 2p + constant$$

Information and Uncertainty

- coin at 50% odds has maximal uncertainty
- reflects my lack of knowledge of the physics
- many ways for 50% heads.
- an election with $p = 0.99$ has a lot of Information

information is the reduction in uncertainty from learning an outcome

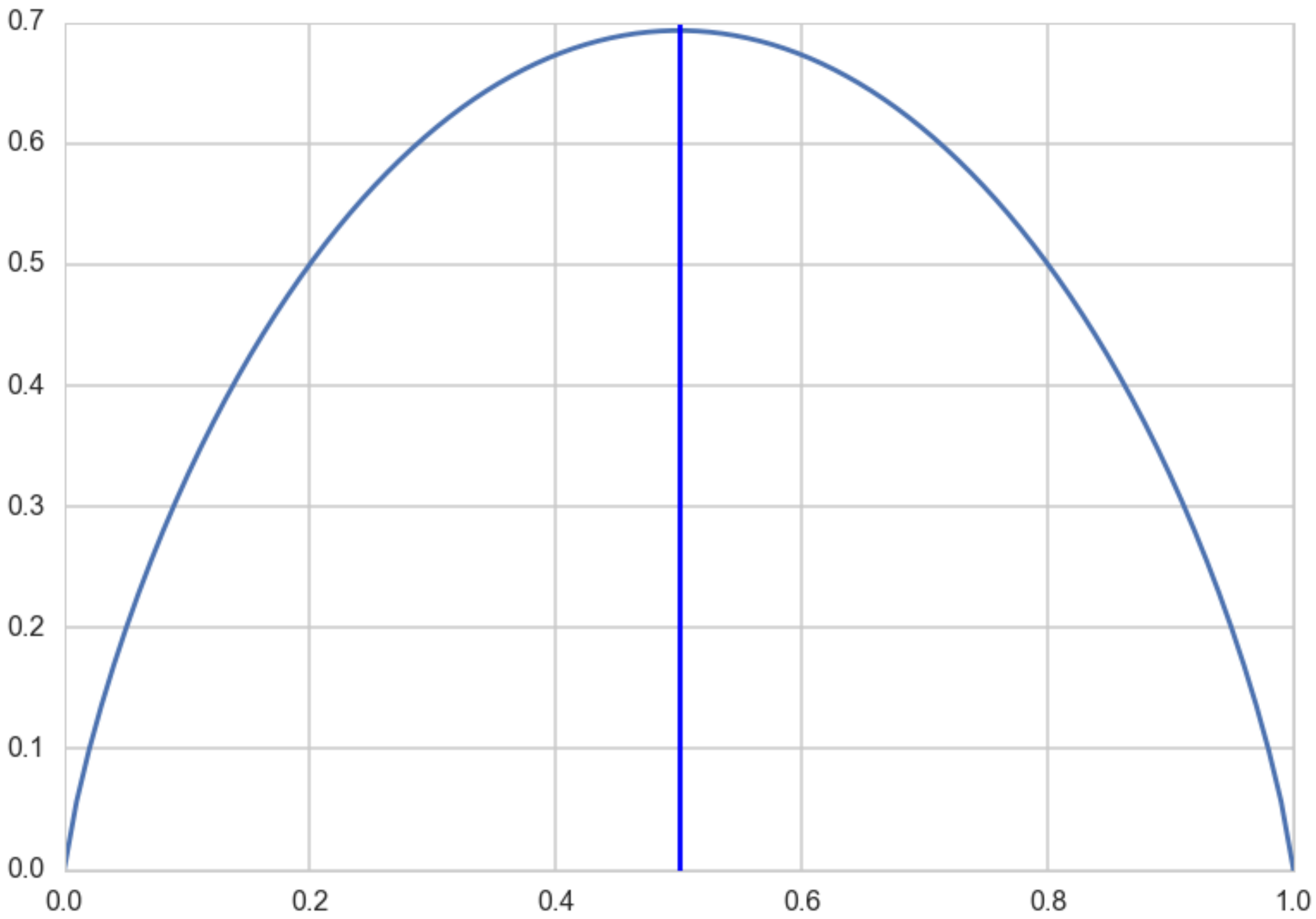
Information Entropy, a measure of uncertainty

Desiderata:

- must be continuous so that there are no jumps
- must be additive across events or states, and must increase as the number of events/states increases

$$H(p) = -E_p[log(p)] = -\int p(x)log(p(x))dx \;\; OR \; -\sum_i p_i log(p_i)$$

Entropy for coin fairness



$$H(p) = -E_p[log(p)] = -p * log(p) - (1-p) * log(1-p)$$

```
def h(p):
    if p==1.:
        ent = 0
    elif p==0.:
        ent = 0
    else:
        ent = - (p*math.log(p) + (1-p)* math.log(1-p))
```

Maximum Entropy (MAXENT)

- finding distributions consistent with constraints and the current state of our information
- what would be the least surprising distribution?
- The one with the least additional assumptions?

The distribution that can happen in the most ways is the one with the highest entropy

For a gaussian

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

$$H(p) = E_p[log(p)] = E_p[-\frac{1}{2}log(2\pi\sigma^2) - (x-\mu)^2/2\sigma^2]$$

$$= -\frac{1}{2}log(2\pi\sigma^2) - \frac{1}{2\sigma^2}E_p[(x-\mu)^2] = -\frac{1}{2}log(2\pi\sigma^2) - \frac{1}{2} = \frac{1}{2}log(2\pi e\sigma^2)$$

Cross Entropy

$$H(p,q) = -E_p[log(q)]$$

Then one can write:

$$D_{KL}(p,q) = H(p,q) - H(p)$$

KL-Divergence is additional entropy introduced by using q instead of p.

We saw this for Logistic regression

- $H(p, q)$ and $D_{KL}(p, q)$ are not symmetric.
- if you use a unusual , low entropy distribution to approximate a usual one, you will be more surprised than if you used a high entropy, many choices one to approximate an unusual one.

Corollary: if we use a high entropy distribution to approximate the true one, we will incur lesser error.

Gaussian is MAXENT for fixed mean and variance

Consider
$D_{KL}(q,p) = E_q[log(q/p)] = H(q,p) - H(q) >= 0$

$$H(q,p) = E_q[log(p)] = -\frac{1}{2}log(2\pi\sigma^2) - \frac{1}{2\sigma^2}E_q[(x-\mu)^2]$$

$E_q[(x-\mu)^2]$ is CONSTRAINED to be σ^2.

$$H(q,p) = -\frac{1}{2}log(2\pi\sigma^2) - \frac{1}{2} = -\frac{1}{2}log(2\pi e\sigma^2) = H(p) >= H(q)!!!$$

Importance of MAXENT

- most common distributions used as likelihoods (and priors) are in the exponential family, MAXENT subject to different constraints.
- gamma: MAXENT all distributions with the same mean and same average logarithm.
- exponential: MAXENT all non-negative continuous distributions with the same average inter-event displacement

Importance of MAXENT

- Information entropy enumerates the number of ways a distribution can arise, after having fixed some assumptions.
- choosing a maxent distribution as a likelihood means that once the constraints has been met, no additional assumptions.

The most conservative distribution

MLE for Logistic Regression

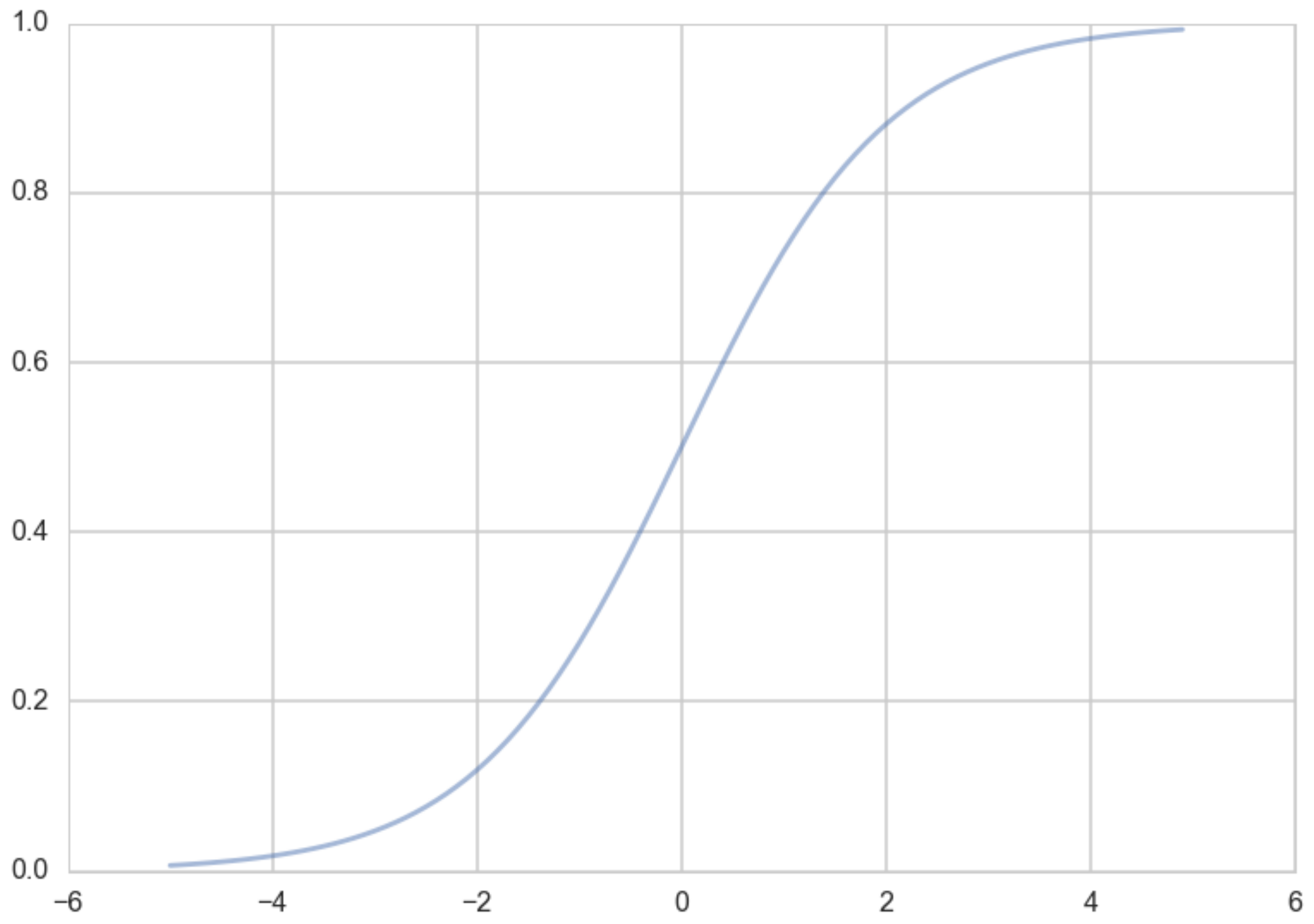
- example of a Generalized Linear Model (GLM)
- "Squeeze" linear regression through a **Sigmoid** function
- this bounds the output to be a probability
- What is the sampling Distribution?

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

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

Use Convex optimization! (soon, hw)