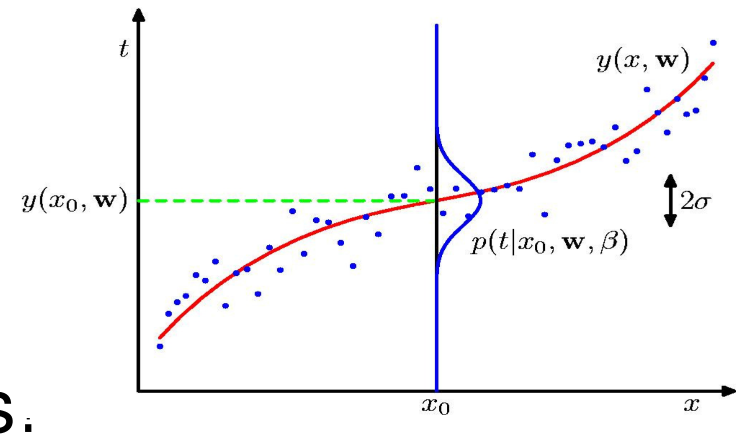
***HTF: Ch3, 7***
*B: Ch3*

# Linear Regression, Regularization Bias-Variance Tradeoff

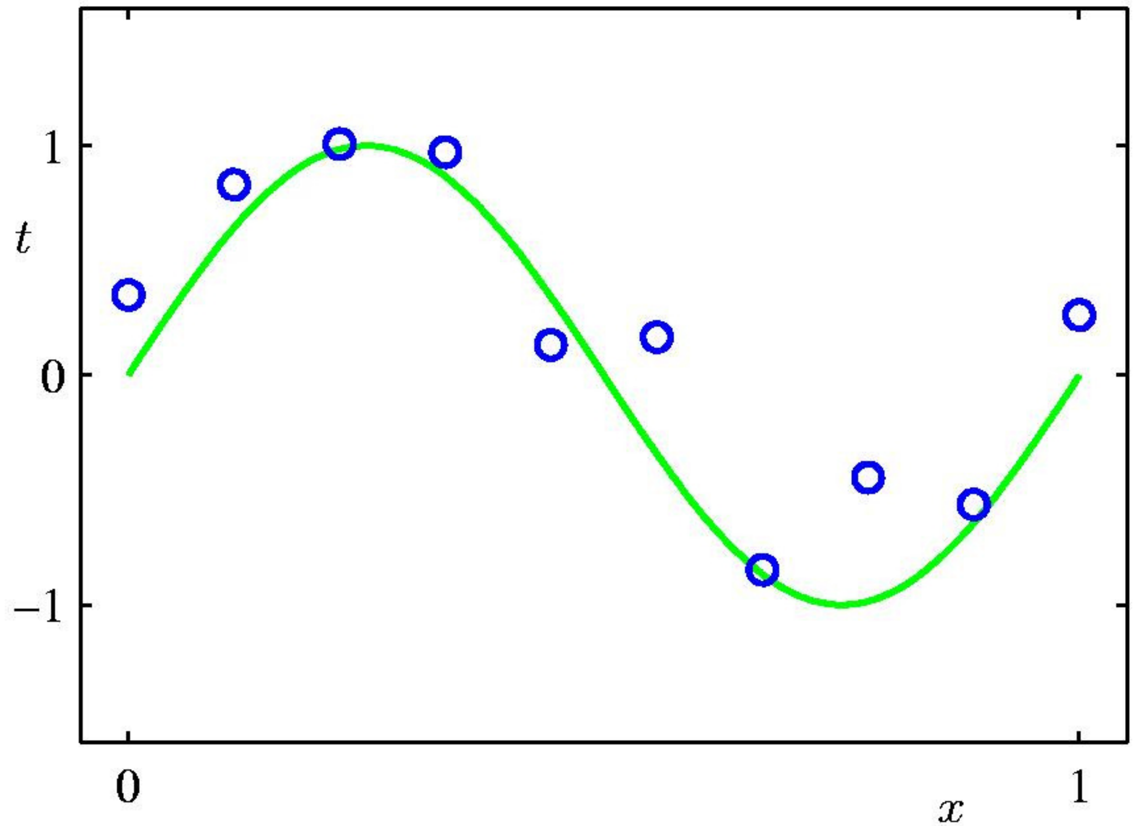
Thanks to C Guestrin, T Dietterich, R Parr, N Ray

# Outline



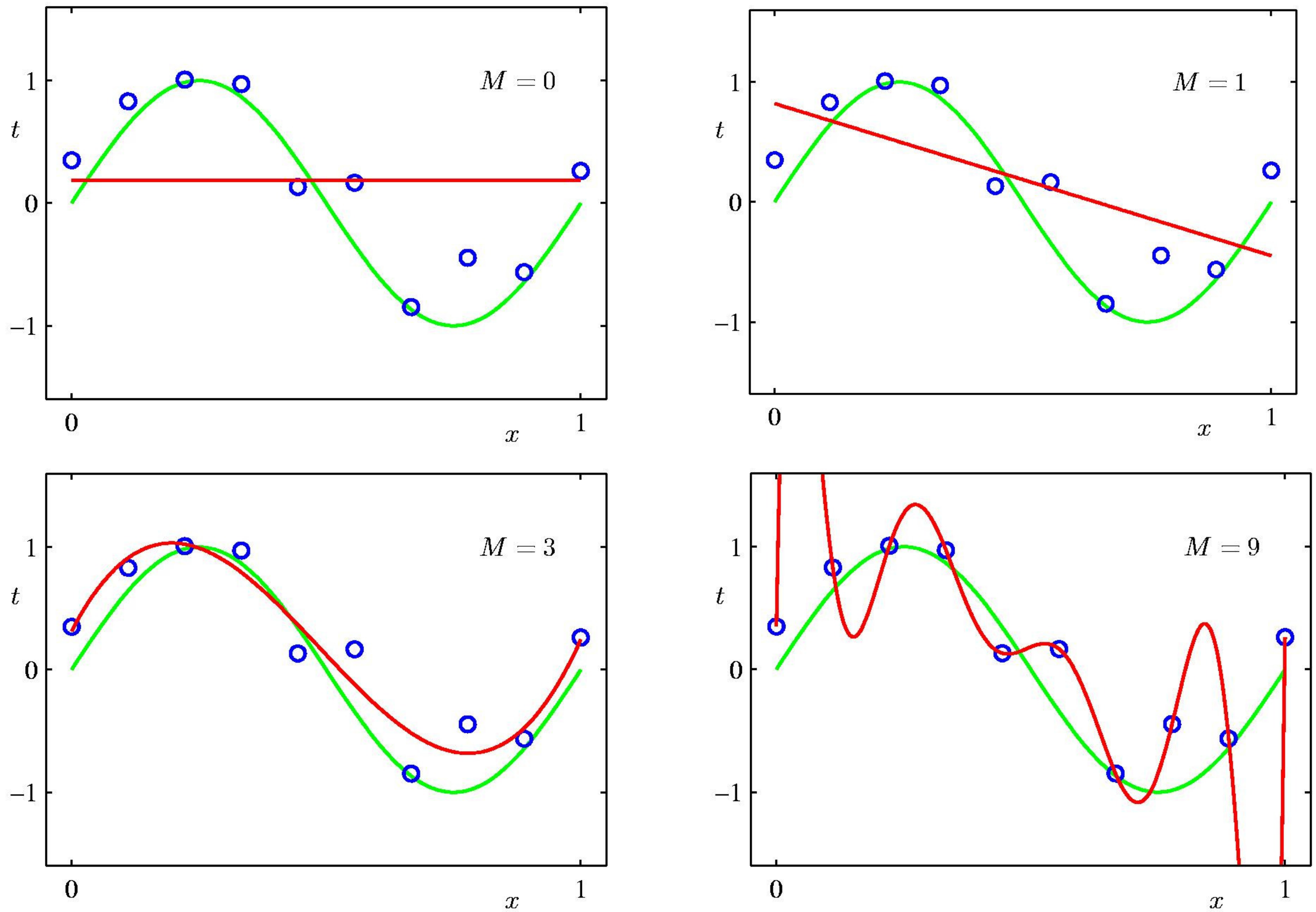

- Linear Regression
  - MLE = Least Squares
  - Basis functions
- Evaluating Predictors
  - Training set error vs Test set error
  - Cross Validation
- Model Selection
  - Bias-Variance analysis
  - Regularization, Bayesian Model

# What is best choice of Polynomial?



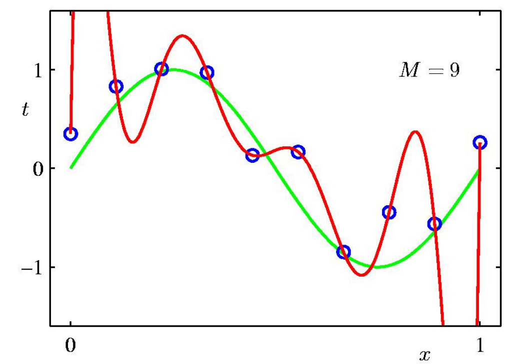

Noisy Source Data

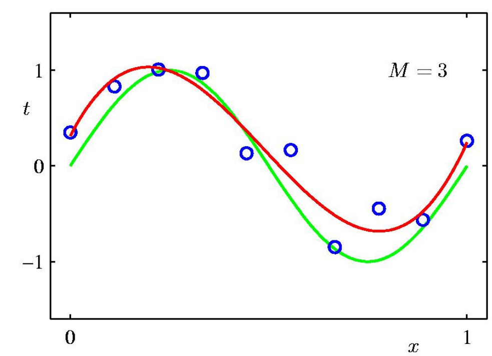
# Fit using Degree 0,1,3,9

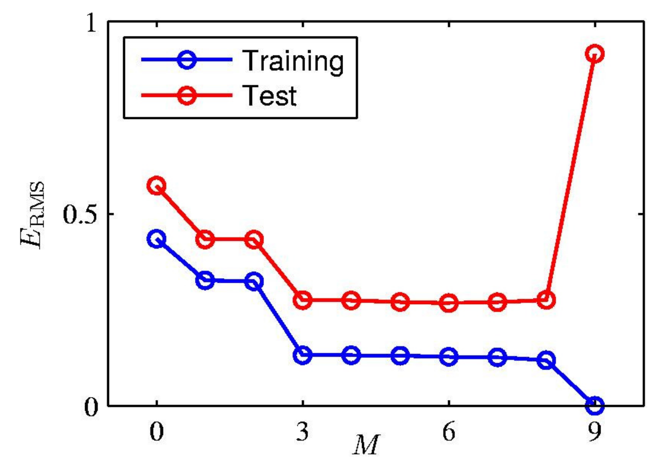


# Comparison

- Degree 9 is the best match to the samples (over-fitting)
- Degree 3 is the best match to the source
- Performance on test data:

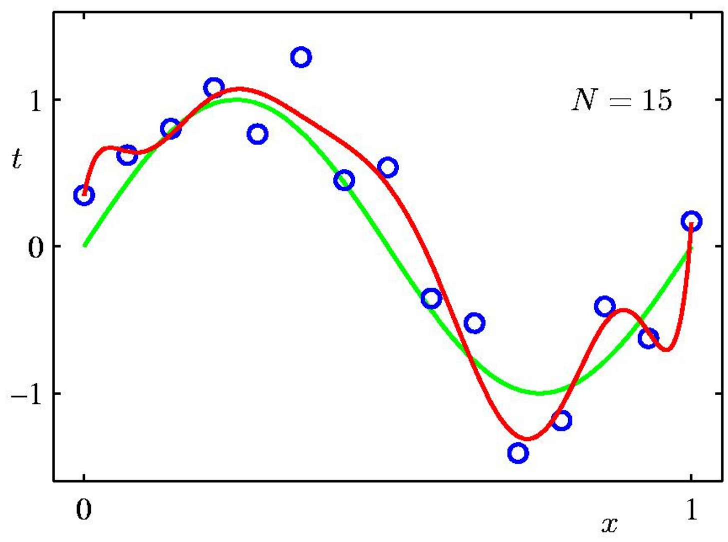


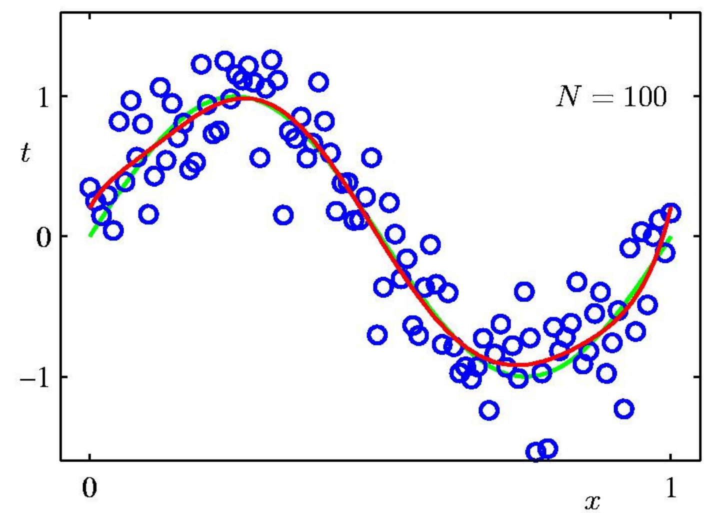



# What went wrong?

- A bad choice of polynomial?
- Not enough data?
  - Yes



# Terms

- $\mathbf{x}$ – input variable
  - $\mathbf{x}^*$ – new input variable
- $h(\mathbf{x})$ – "truth" – underlying response function
- $t = h(\mathbf{x}) + \varepsilon$ – actual observed response
- $y(\mathbf{x}; D)$ – predicted response, based on model learned from dataset D
- $\hat{y}(\mathbf{x}) = E_D[\ y(\mathbf{x}; D)\ ]$ – expected response, averaged over (models based on) all datasets
- $Eerr = E_{D,(x^*,t^*)}[\ (t^* - y(\mathbf{x}^*))^2\ ]$ – expected $L_2$ error on new instance $\mathbf{x}^*$

# Bias-Variance Analysis in Regression

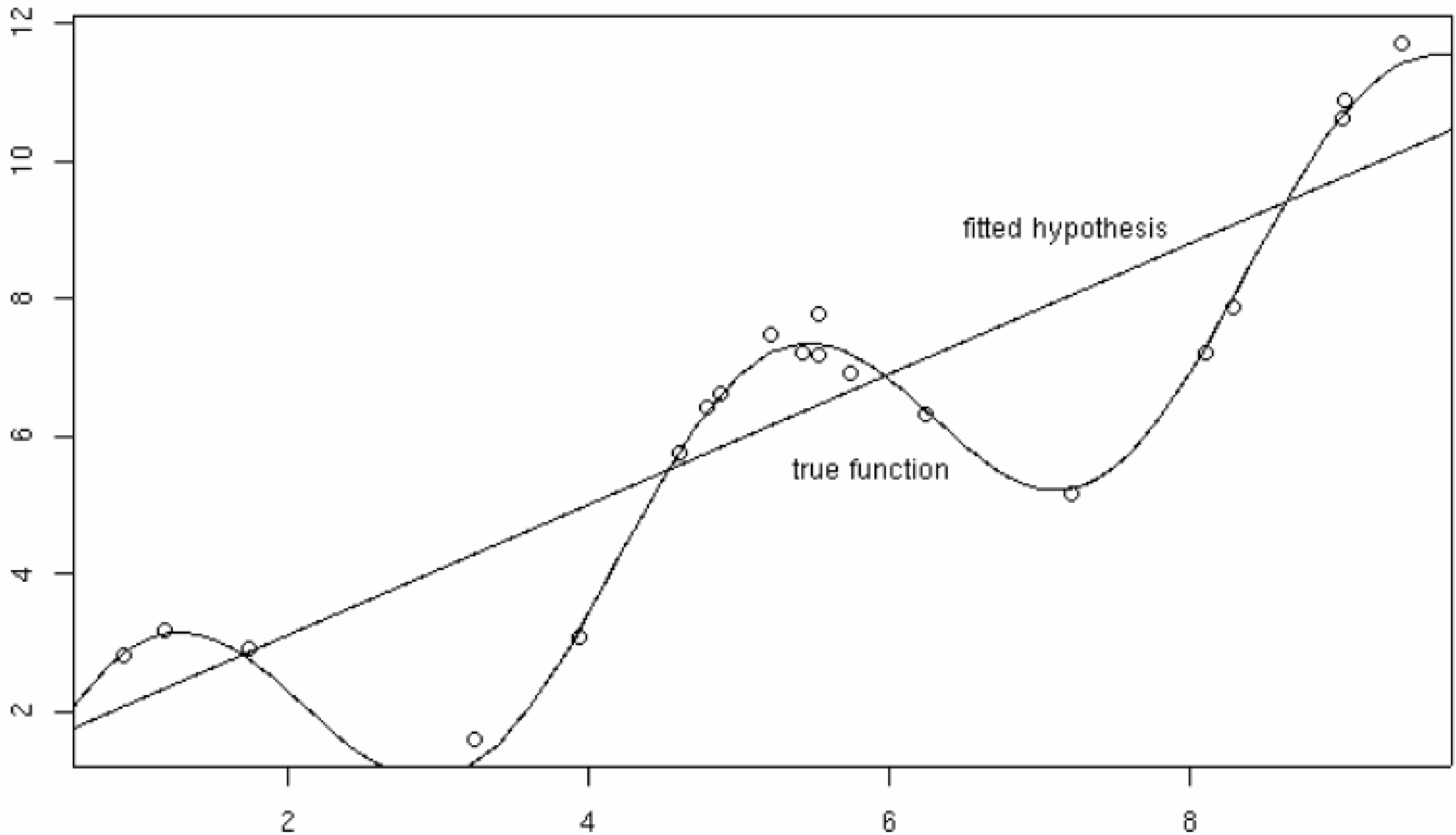
- Observed value is $t(\mathbf{x}) = h(\mathbf{x}) + \varepsilon$
  - $\varepsilon \sim N(0, \sigma^2)$
    - normally distributed: mean 0, std deviation $\sigma^2$
  - Note: $h(\mathbf{x}) = E[\, t(x) \mid \mathbf{x} \,]$
- Given training examples, $D = \{(\mathbf{x}_i, t_i)\}$, let

  $y(.) = y(.; D)$

  be predicted function,
  based on model learned using D
    - Eg, linear model $y_w(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0$ using $\mathbf{w}$ =MLE(D)

# Example: 20 points

$$t = x + 2\sin(1.5x) + \mathcal{N}(0, 0.2)$$



# Bias-Variance Analysis

- Given a *new* data point $\mathbf{x}^*$
  - return predicted response: $y(\mathbf{x}^*)$
  - observed response: $t^* = h(\mathbf{x}^*) + \varepsilon$
- The *expected prediction error* is …

$$\mathrm{Eerr} = E_{D,(x^*,t^*)}[\ (t^* - y(\mathbf{x}^*))^2\ ]$$

# Expected Loss

- $[y(\mathbf{x}) - t]^2 = [y(\mathbf{x}) - h(\mathbf{x}) + h(\mathbf{x}) - t]^2 =$
  $[y(\mathbf{x}) - h(\mathbf{x})]^2$
  $+ 2\,[y(\mathbf{x}) - h(\mathbf{x})]\,[h(\mathbf{x}) - t]$
  $+ [h(\mathbf{x}) - t]^2$

Expected value is 0 as $h(\mathbf{x}) = E[t|\mathbf{x}]$

- $Eerr = \int [y(\mathbf{x}) - t]^2\, p(\mathbf{x},t)\; d\mathbf{x}\, dt$

$$= \underbrace{\int \{y(\mathbf{x}) - h(\mathbf{x})\}^2\, p(\mathbf{x})\, d\mathbf{x}}_{} \;+\; \underbrace{\int \{h(\mathbf{x}) - t\}^2\, p(\mathbf{x}, t)\, d\mathbf{x}\, dt}_{}$$

Mismatch between OUR hypothesis y(.) & target h(.)
… we can influence this

Noise in distribution of target
… nothing we can do

$$Eerr = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2\, p(\mathbf{x})\, d\mathbf{x} \;+\; \cancel{\int \{h(\mathbf{x}) - t\}^2\, p(\mathbf{x},t)\, d\mathbf{x}\, dt}$$

# Relevant Part of Loss

- Really $y(\mathbf{x}) = y(\mathbf{x}; D)$ fit to data $D$…
  so consider *expectation over data sets* $D$
  - Let $\hat{y}(\mathbf{x}) = E_D[y(\mathbf{x}; D)]$
- $E_D[\, \{h(\mathbf{x}) - y(\mathbf{x}; D)\}^2 \,]$

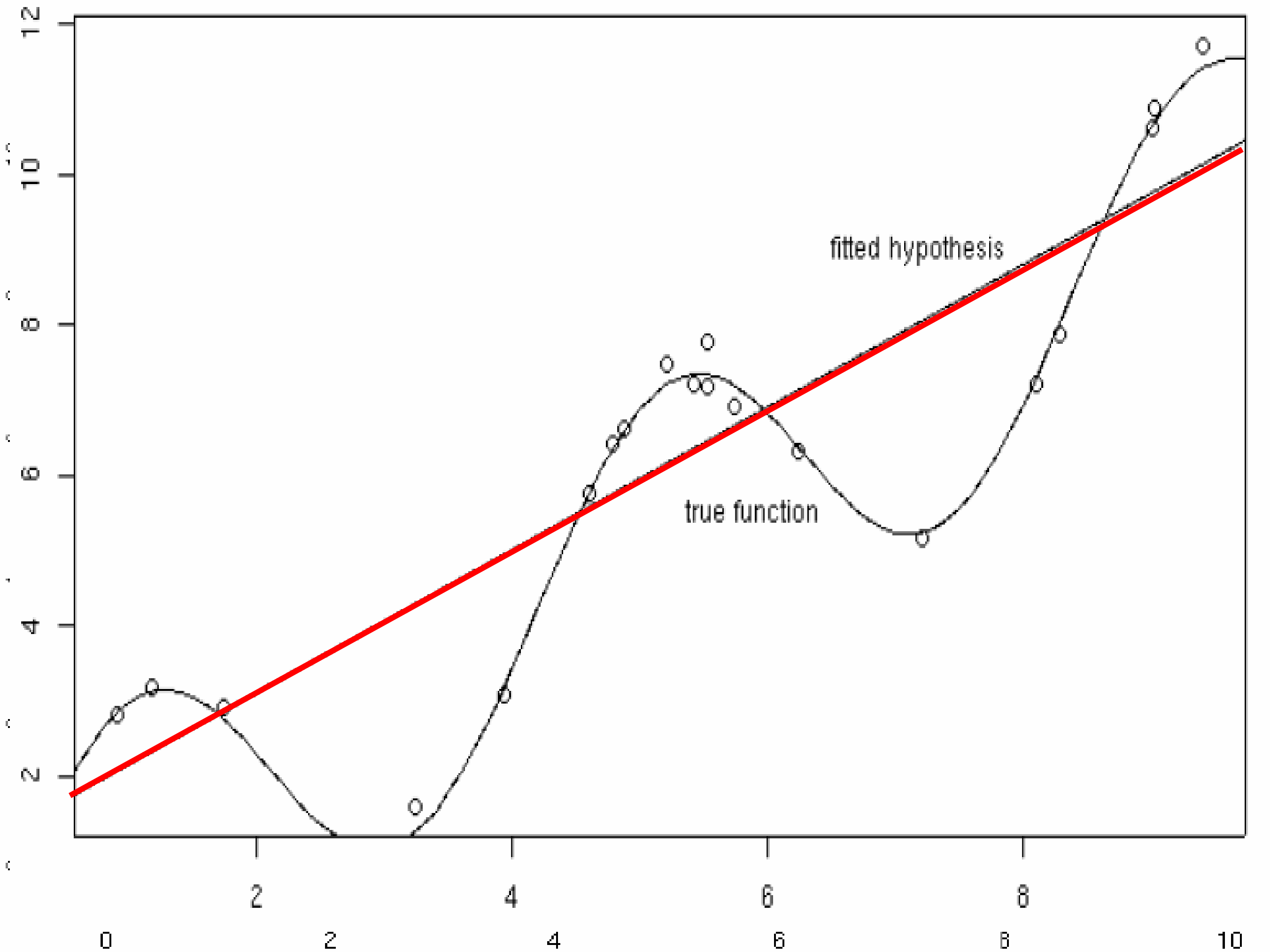
$$= E_D[h(\mathbf{x}) - \hat{y}(x) + \hat{y}(x) - y(\mathbf{x}; D)\,]\}^2$$

$$= E_D[\, \{h(\mathbf{x}) - \hat{y}(x)\}^2] + \underbrace{\cancel{2E_D[\, \{h(\mathbf{x}) - \hat{y}(x)\}\, \{\, y(\mathbf{x}; D) - E_D[y(\mathbf{x}; D)\,\}]}}_{0}$$

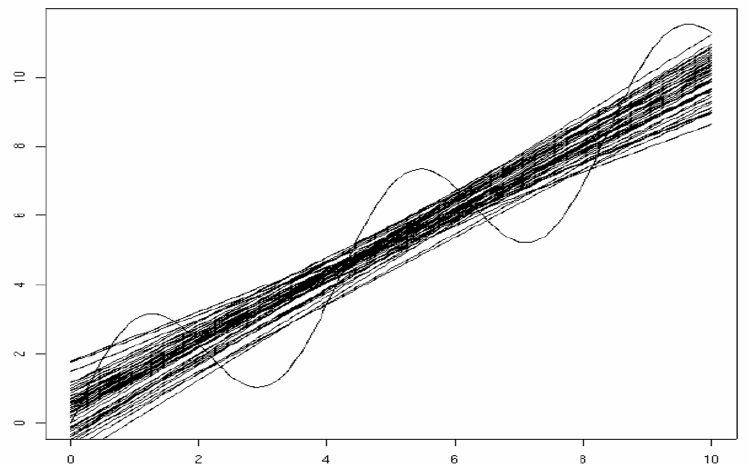
$$+ E_D[\{\, y(\mathbf{x}; D) - E_D[y(\mathbf{x}; D)]\,\}^2\,]$$

$$= \underbrace{\{h(\mathbf{x}) - \hat{y}(\mathbf{x})\}^2}_{\text{Bias}^2} + \underbrace{E_D[\, \{\, y(\mathbf{x}; D) - \hat{y}(\mathbf{x})\,\}^2\,]}_{\text{Variance}}$$

# 50 fits (20 examples each)

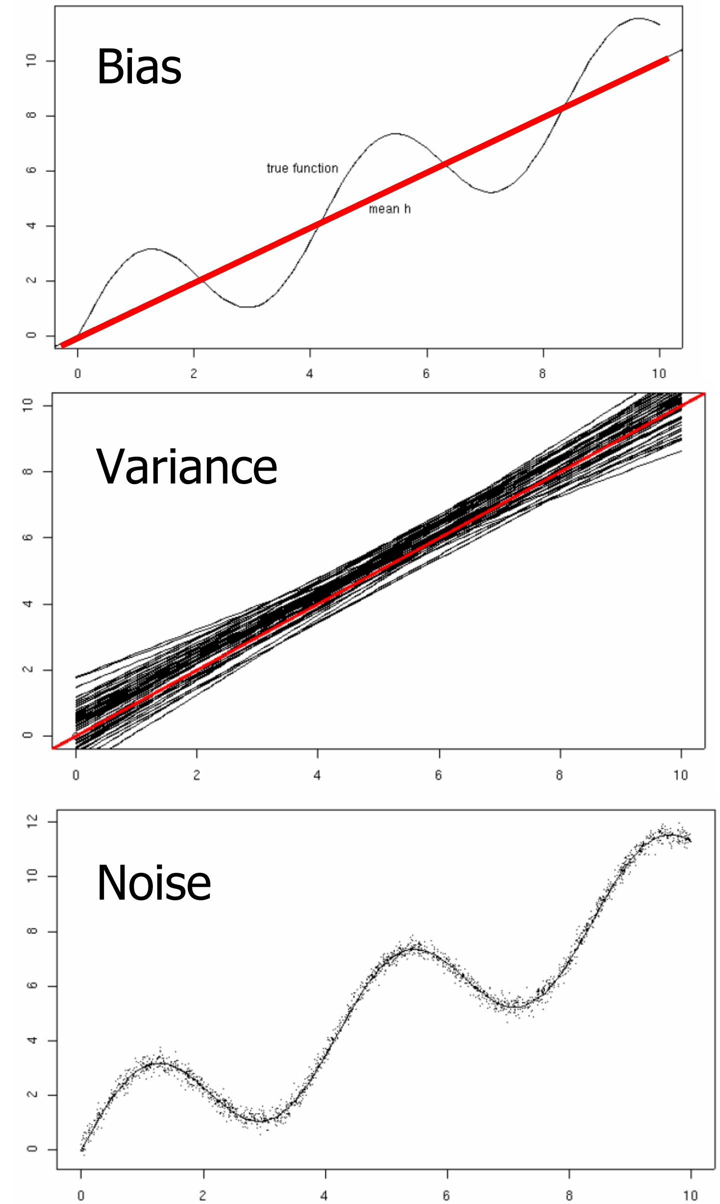


# Bias, Variance, Noise

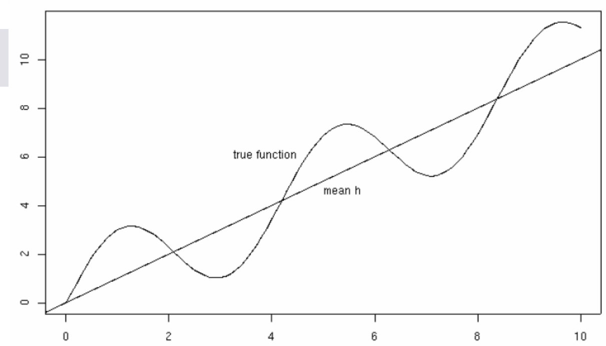


50 fits (20 examples each)

=

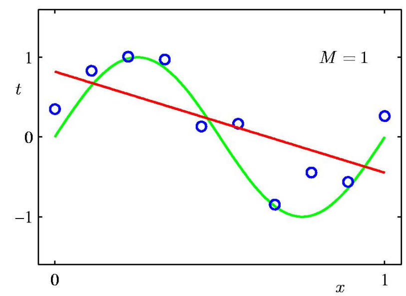


# Understanding Bias



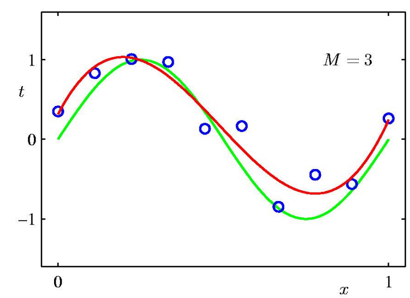

$$\{\ \hat{y}(\mathbf{x}) - h(\mathbf{x})\ \}^2$$

- Measures how well *our approximation architecture* can fit the data




- Weak approximators
  - (e.g. low degree polynomials)

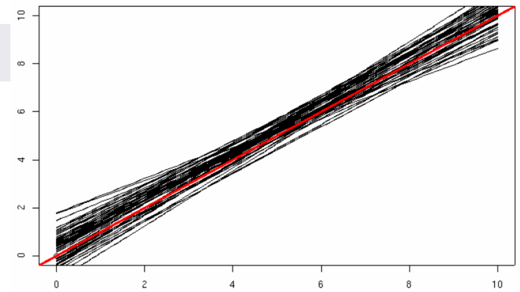
  will have high bias




- Strong approximators
  - (e.g. high degree polynomials)

  will have lower bias

# Understanding Variance



$$\boxed{E_D[\ \{\ y(\mathbf{x}; D) - \hat{y}_D(\mathbf{x})\ \}^2\ ]}$$

- No *direct* dependence on target values
- For a fixed size D:
  - **Strong** approximators tend to have **more variance** … different datasets will lead to DIFFERENT predictors
  - **Weak** approximators tend to have **less variance** … slightly different datasets may lead to SIMILAR predictors
- Variance will typically disappear as $|D| \to \infty$

# Summary of Bias,Variance,Noise

- Eerr $=$ $E[\,(t^* - y(\mathbf{x}^*))^2\,]$ $=$

  $E[\,(y(\mathbf{x}^*) - \hat{y}(\mathbf{x}^*))^2\,]$

  $+\ (\hat{y}(\mathbf{x}^*) - h(\mathbf{x}^*))^2$

  $+\ E[\,(t^* - h(\mathbf{x}^*))^2\,]$

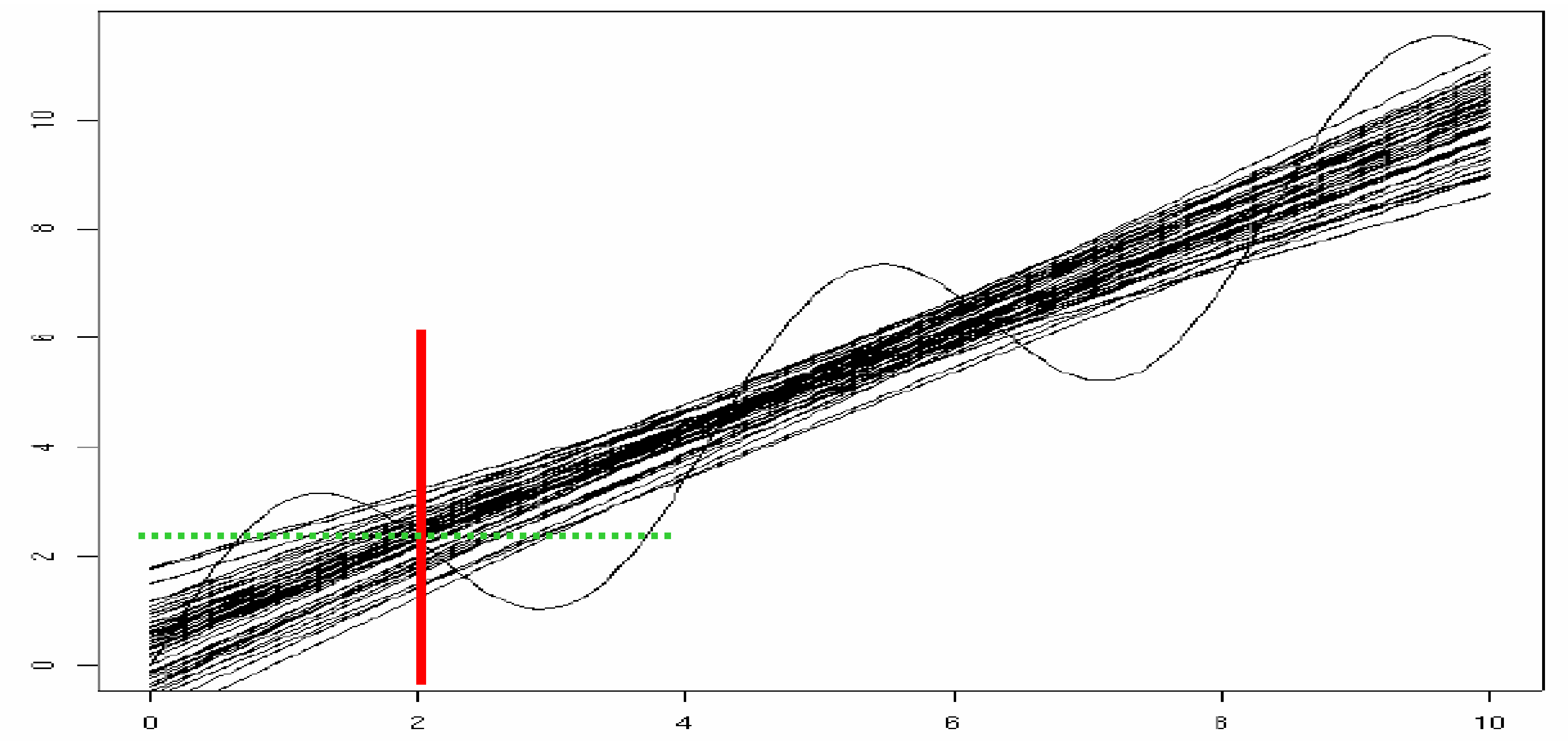
  $=$ Var( h(x*) ) + Bias( h(x*) )$^2$ + Noise

Expected prediction error
= Variance + Bias$^2$ + Noise

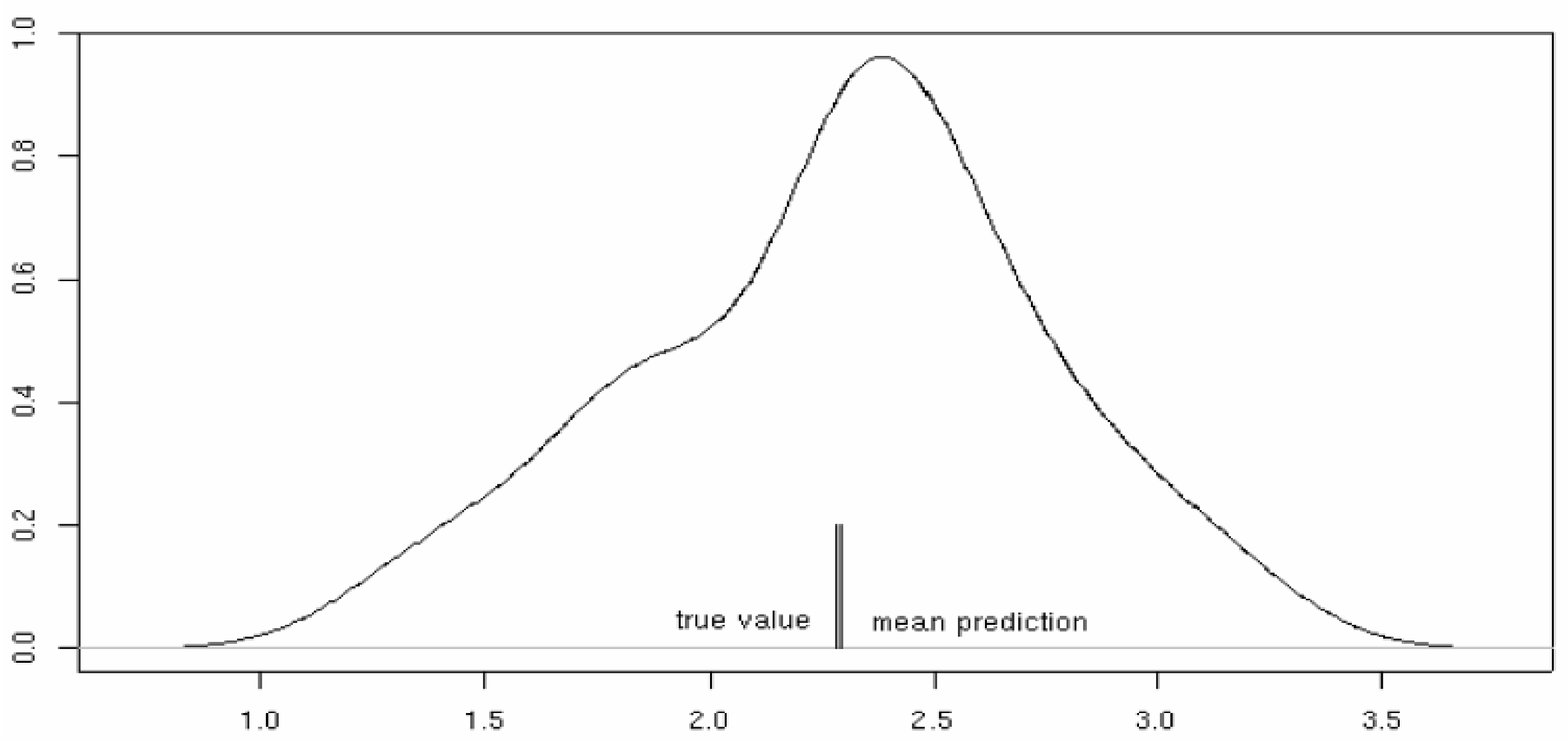
# Bias, Variance, and Noise

- **Bias**: $\hat{y}(\mathbf{x}^*) - h(\mathbf{x}^*)$
  - the best error of model $\hat{y}(x^*)$ [average over datasets]
- **Variance**: $E_D[\,(\,y_D(\mathbf{x}^*) - \hat{y}(\mathbf{x}^*)\,)^2\,]$
  - How much $y_D(x^*)$ varies from one training set $D$ to another
- **Noise**: $E[\,(t^* - h(\mathbf{x}^*))^2\,] = E[\varepsilon^2] = \sigma^2$
  - How much $t^*$ varies from $h(\mathbf{x}^*) = t^* + \varepsilon$
  - Error, even given PERFECT model, and $\infty$ data

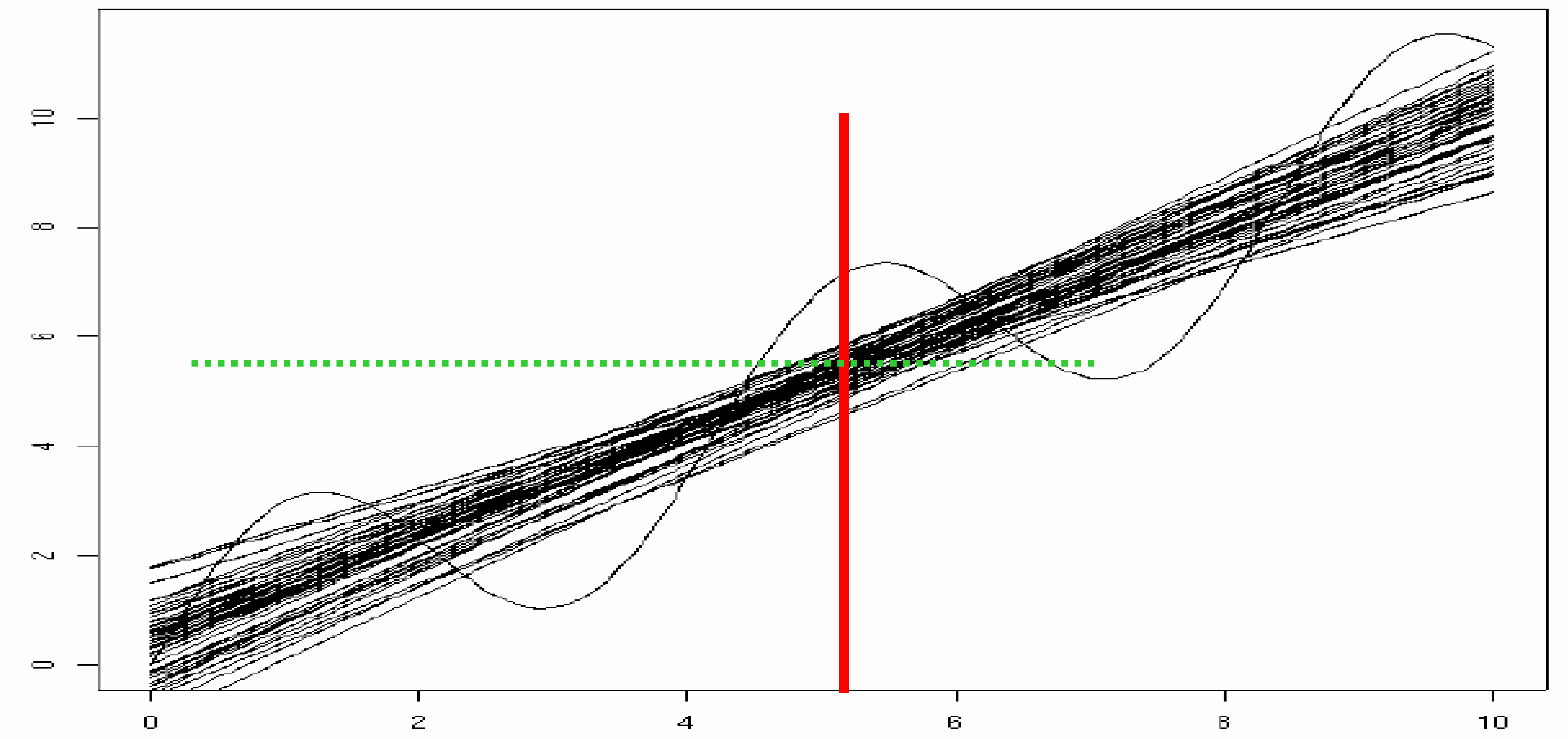
# 50 fits (20 examples each)



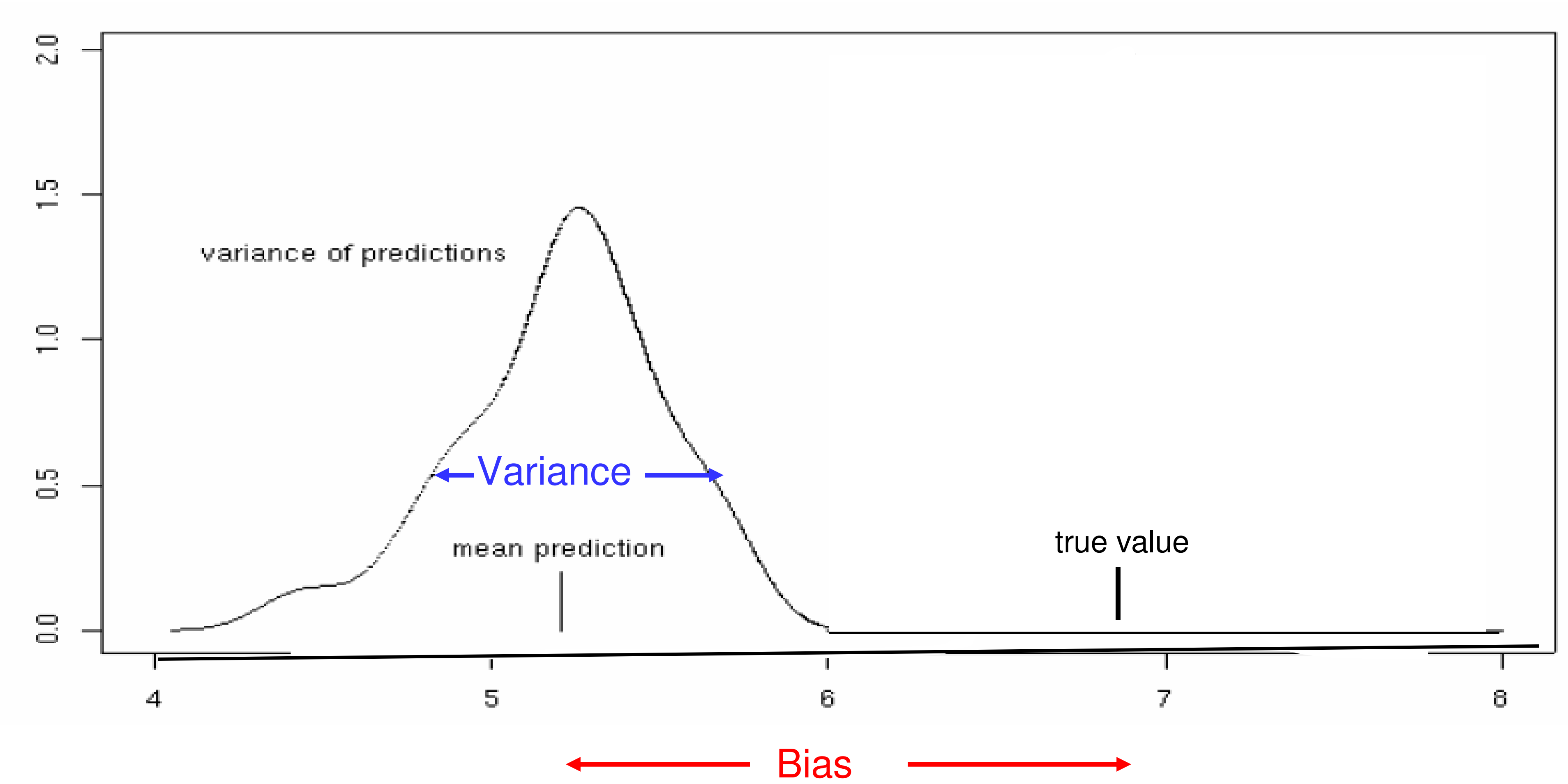
# Predictions at x=2.0



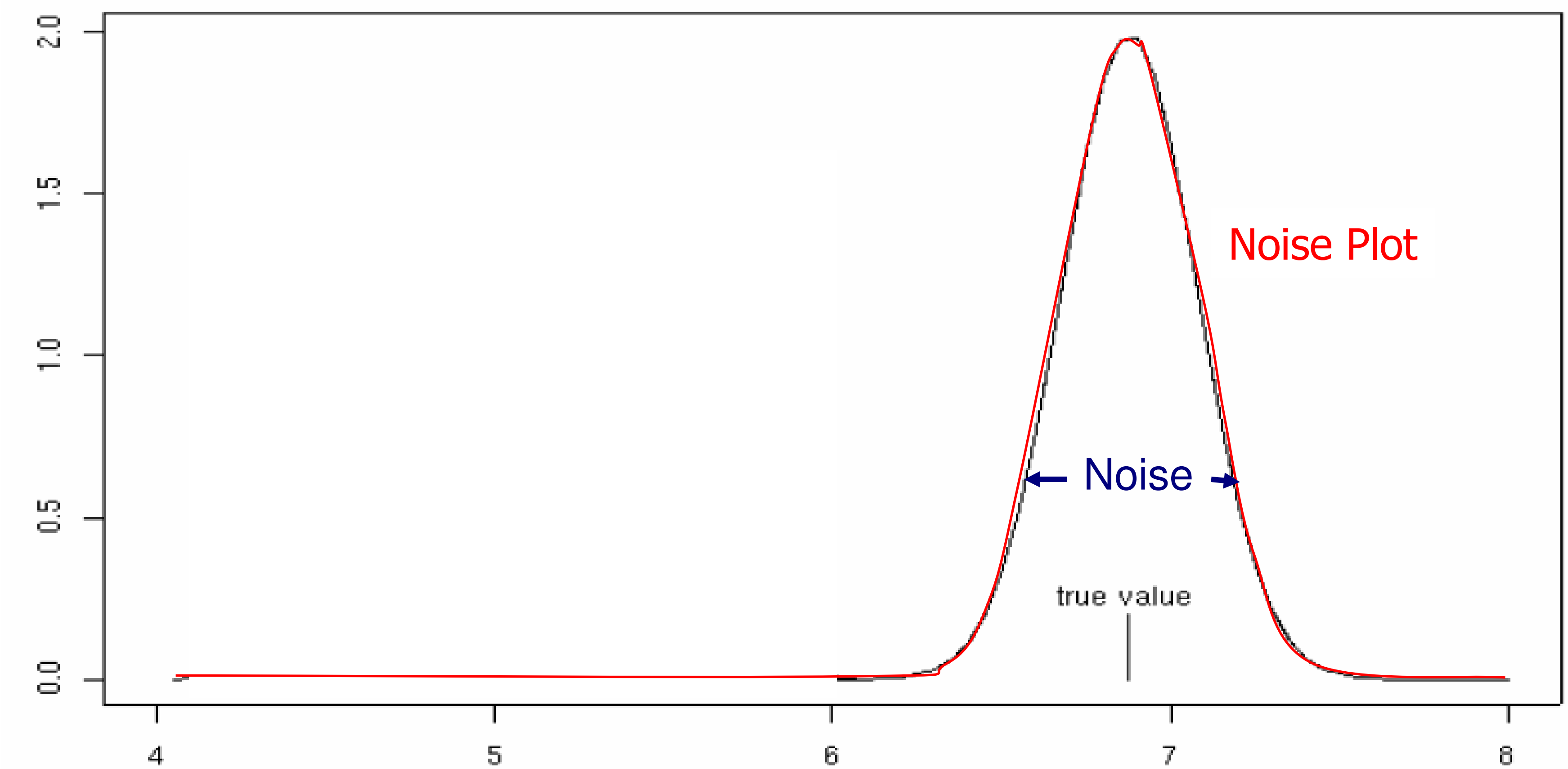
# 50 fits (20 examples each)



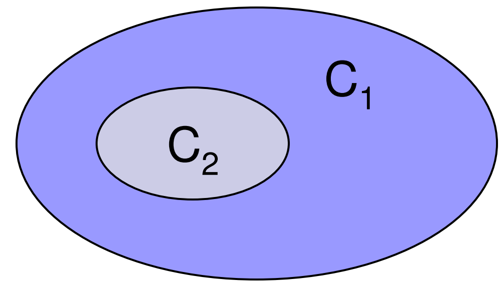
# Predictions at x=5.0



# Observed Responses at x=5.0



# Model Selection: Bias-Variance

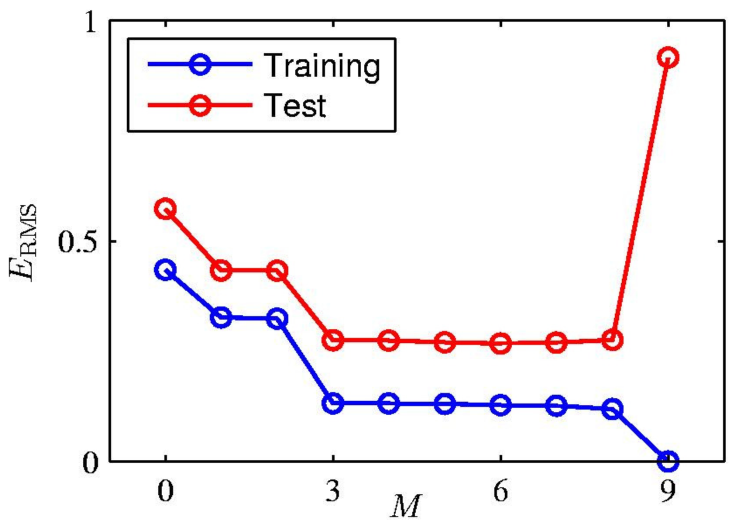


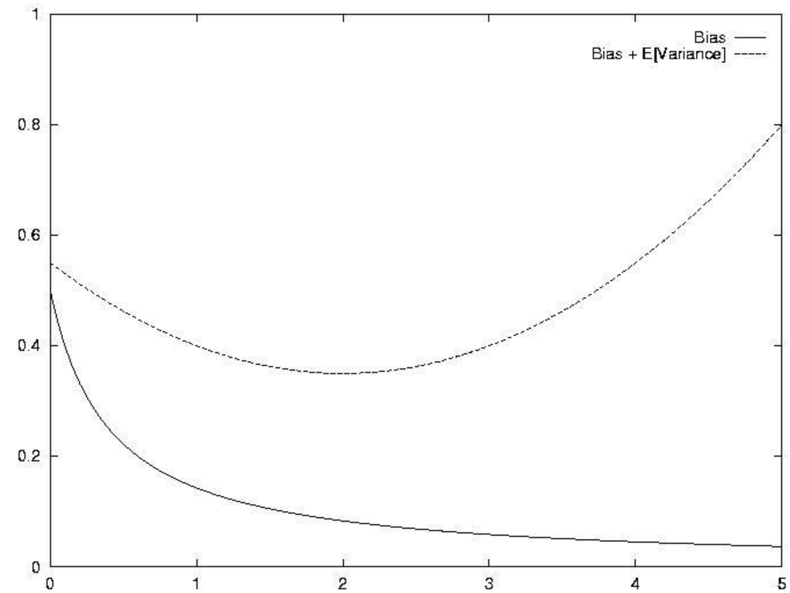

- $C_1$ "more expressive than" $C_2$
  iff
  representable in $C_1 \Rightarrow$ representable in $C_2$
  "$C_2 \subset C_1$"
- Eg, LinearFns $\subset$ QuadraticFns
  0-HiddenLayerNNs $\subset$ 1-HiddenLayerNNs

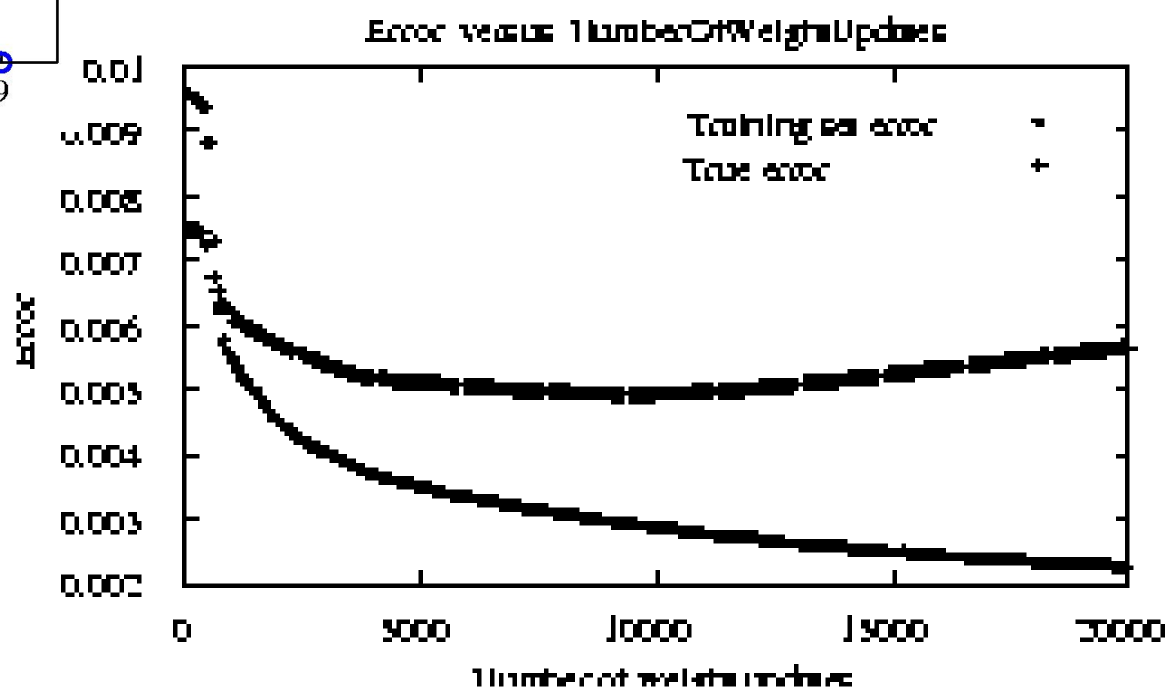
$\Rightarrow$ can ALWAYs get better fit using $C_1$, over $C_2$

- But … sometimes better to look for $y \in C_2$

# Standard Plots..



# Why?

- $C_2 \subset C_1 \Rightarrow$
  $\forall\, y \in C_2$
  $\exists\, x^* \in C_1$ that is at-least-as-good-as y

- But given *limited sample,*
  might not find this best $x^*$

- Approach: consider $\text{Bias}^2 + \text{Variance}$!!

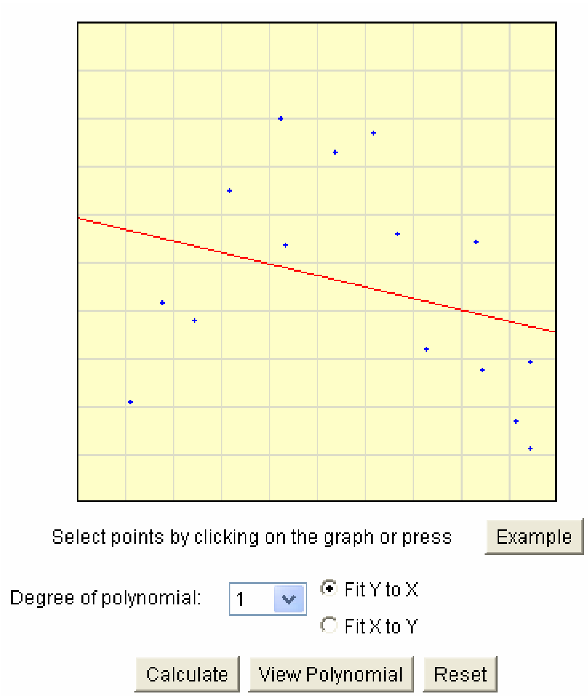
# Bias-Variance tradeoff – Intuition

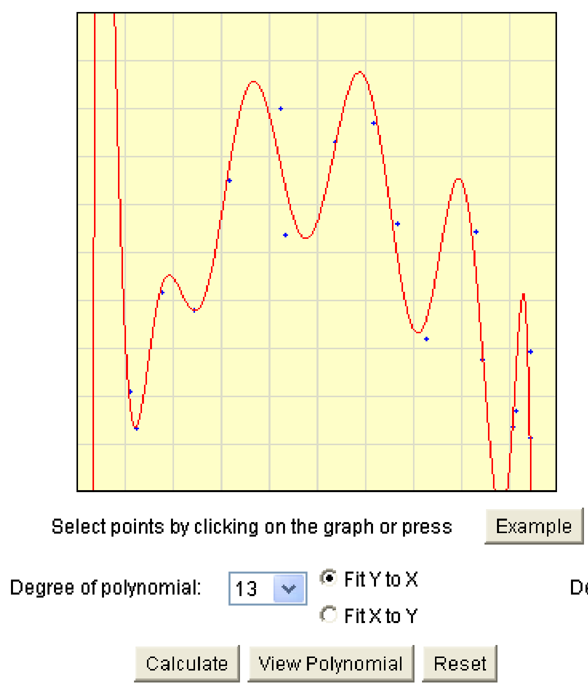
- Model too "simple" $\Rightarrow$ does *not* fit the data well
  - A biased solution

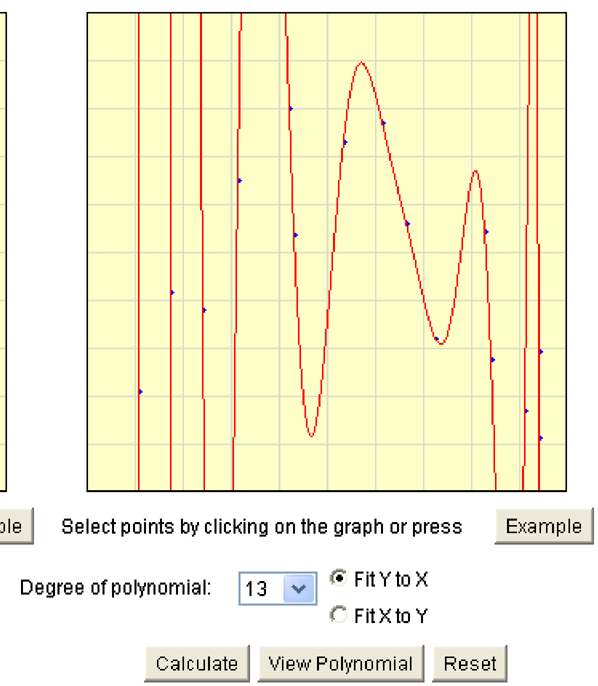
- Model too complex $\Rightarrow$ small changes to the data, changes predictor a lot
  - A high-variance solution

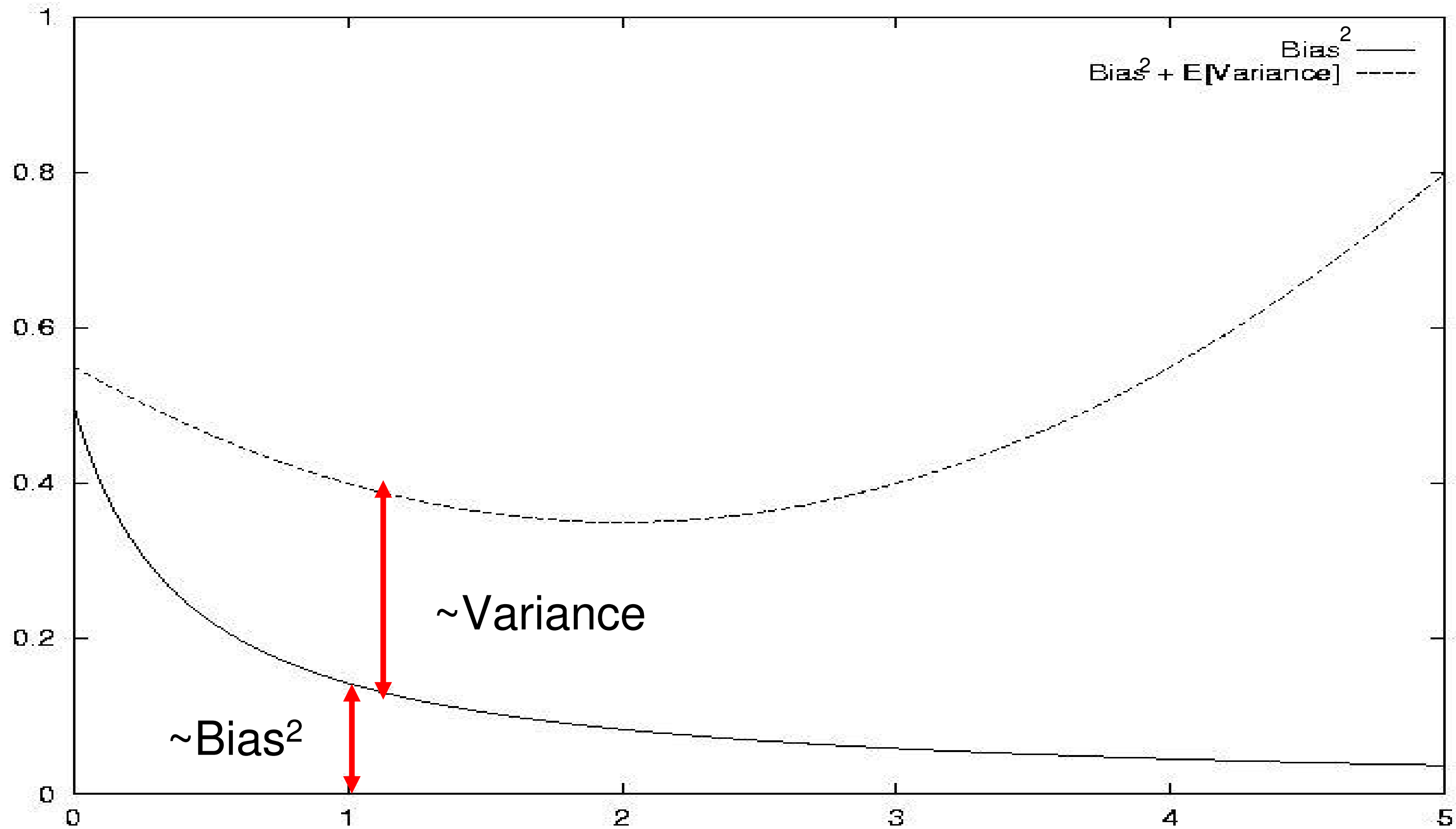
# Bias-Variance Tradeoff

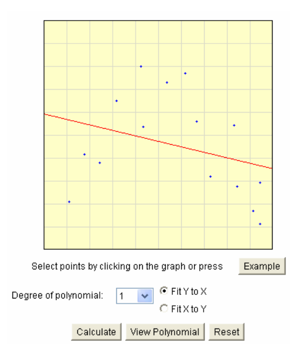
- Choice of hypothesis class introduces learning bias
  - More complex class ⇒ less bias
  - More complex class ⇒ more variance

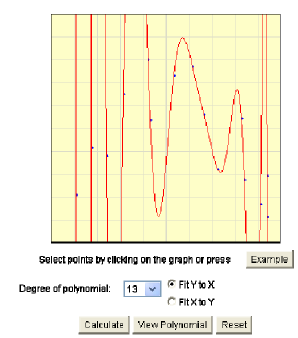


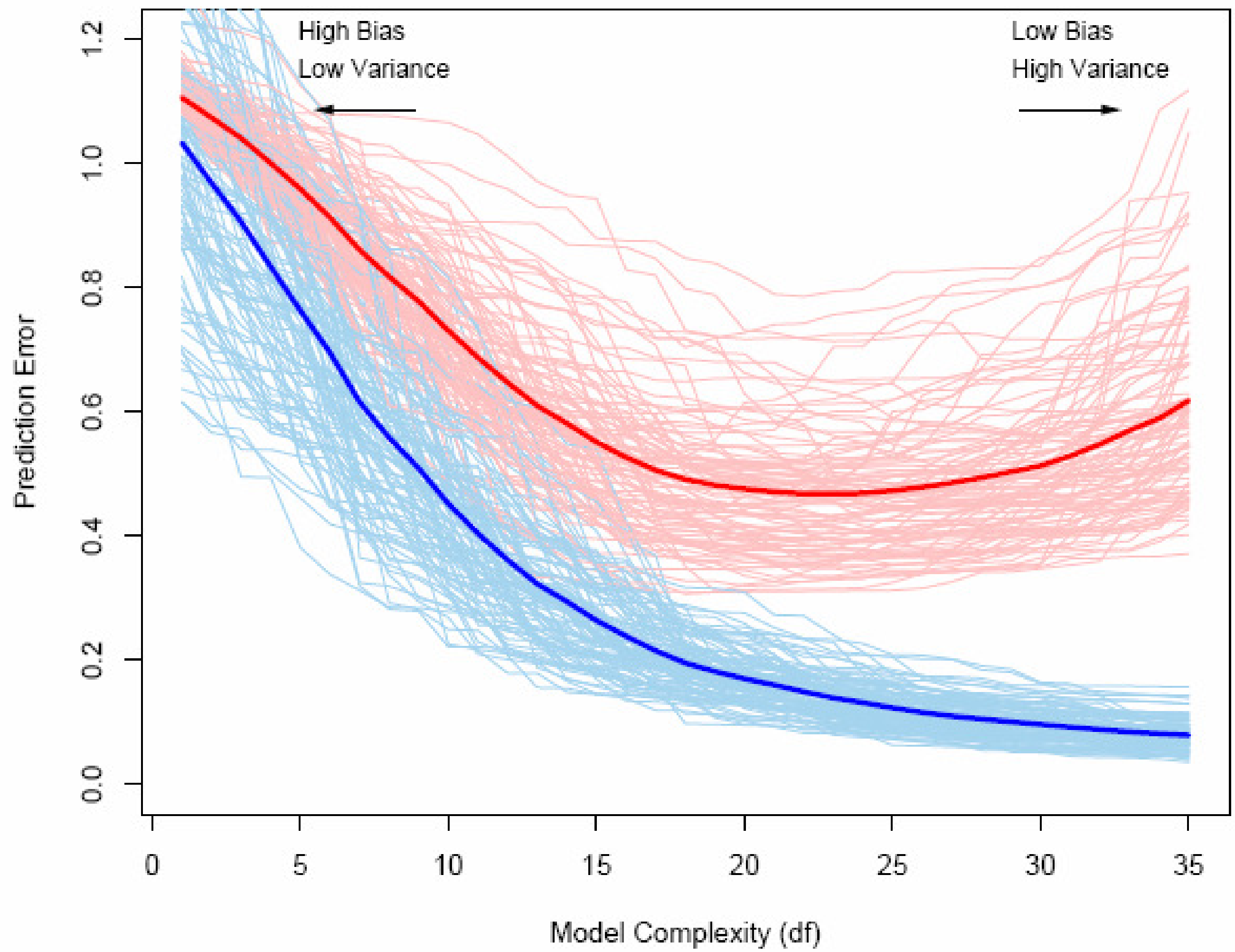




Bias$^2$
Bias$^2$ + E[Variance]
~Variance
~Bias$^2$
0
0.2
0.4
0.6
0.8
1
0
1
2
3
4
5


Select points by clicking on the graph or press
Example
Degree of polynomial:
1
Fit Y to X
Fit X to Y
Calculate
View Polynomial
Reset

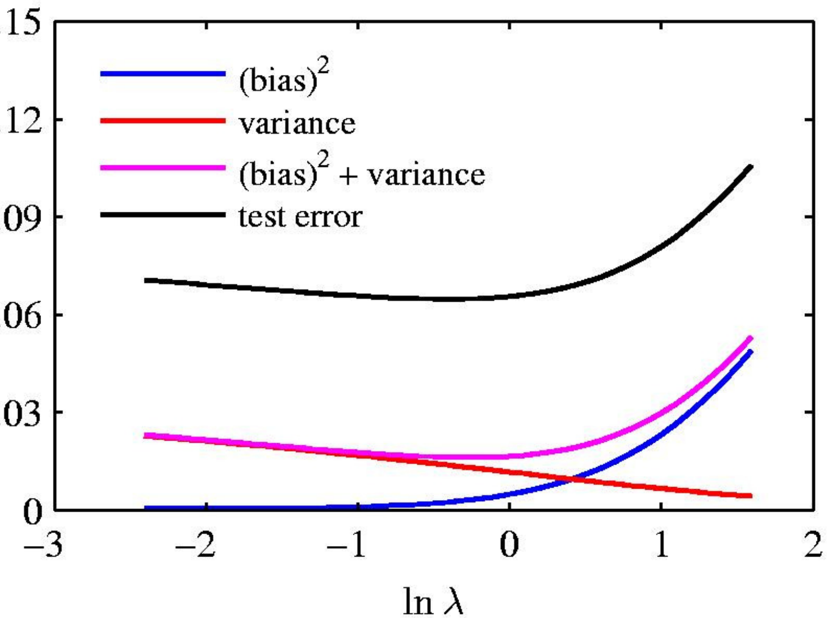

Select points by clicking on the graph or press
Example
Degree of polynomial:
13
Fit Y to X
Fit X to Y
Calculate
View Polynomial
Reset




- *Behavior of test sample and training sample error as function of model complexity*
  - *light blue curves show the training error* err,
  - *light red curves show the conditional test error* Err $T$

  *for* 100 *training sets of size* 50 *each*
- *Solid curves = expected test error* Err *and expected training error* E[err].

# Empirical Study




- Based on different regularizers

# Effect of Algorithm Parameters on Bias and Variance

- k-nearest neighbor:
  - increasing k typically
    increases bias and reduces variance
- decision trees of depth D:
  - increasing D typically
    increases variance and reduces bias
- RBF SVM with parameter $\sigma$:
  - increasing $\sigma$ typically
    increases bias and reduces variance

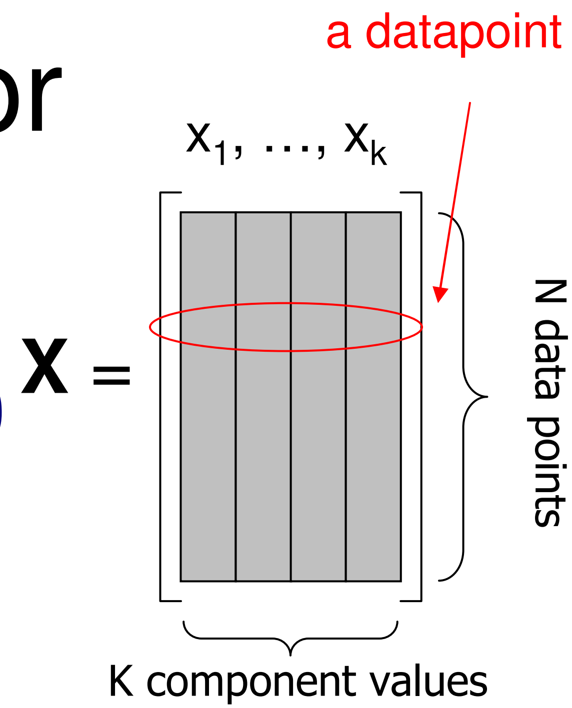
# Least Squares Estimator

- Truth: $f(x) = x^T\beta$

  Observed: $y = f(x) + \varepsilon \quad E[\,\varepsilon\,] = 0$
- Least squares estimator

  $\ell(x_0) = x_0^T\underline{\beta} \quad \underline{\beta} = (X^TX)^{-1}X^T\mathbf{y}$
  - Unbiased: $f(x_0) = E[\,\ell(x_0)\,]$

  $$f(x_0) - E[\,\ell(x_0)\,]$$
  $$= x_0^T\beta - E[\,x_0^T(X^TX)^{-1}X^T\mathbf{y}\,]$$
  $$= x_0^T\beta - E[\,x_0^T(X^TX)^{-1}X^T(X\beta + \varepsilon)\,]$$
  $$= x_0^T\beta - E[\,x_0^T\beta + x_0^T(X^TX)^{-1}X^T\varepsilon\,]$$
  $$= x_0^T\beta - x_0^T\beta + x_0^T(X^TX)^{-1}X^T E[\varepsilon\,] = 0$$



# Gauss-Markov Theorem

- Least squares estimator $\ell(x_0) = x_0^T (X^TX)^{-1}X^Ty$
  - … is unbiased: $f(x_0) = E[\, \ell(x_0) \,]$
  - … is linear in **y** … $\ell(x_0) = c_0^Ty$ where $c_0^T$
- Gauss-Markov Theorem:
  *Least square estimate* has the minimum variance among all linear unbiased estimators.
  - BLUE: Best Linear Unbiased Estimator
- Interpretation: Let $g(x_0)$ be any other …
  - unbiased estimator of $f(x_0)$ … ie, $E[\, g(x_0) \,] = f(x_0)$
  - that is linear in **y** … ie, $g(x_0) = c^T\mathbf{y}$

then $Var[\, \ell(x_0) \,] \leq Var[\, g(x_0) \,]$

# Variance of Least Squares Estimator

- Least squares estimator

  $\ell(x_0) = x_0^T \underline{\beta} \quad \underline{\beta} = (X^TX)^{-1}X^T\mathbf{y}$

- Variance:

  $E[\ (\ell(x_0) - E[\ \ell(x_0)\ ]\ )^2\ ]$

  $= E[\ (\ell(x_0) - f(x_0)\ )^2\ ]$

  $= E[\ (\ x_0^T\ (X^TX)^{-1}X^T\ \underline{\beta} - x_0^T\beta\ )^2\ ]$

  $= E[\ \ (x_0^T(X^TX)^{-1}X^T(X\beta + \varepsilon) - x_0^T\beta\ )^2\ ]$

  $= E[\ \ (x_0^T\beta + x_0^T(X^TX)^{-1}X^T\varepsilon - x_0^T\beta\ )^2\ ]$

  $= E[\ \ (x_0^T(X^TX)^{-1}X^T\varepsilon)^2\ ]$

  $= \sigma_\varepsilon^2\ p/N$

$y = f(x) + \varepsilon \quad E[\ \varepsilon\ ] = 0$

$var(\varepsilon) = \sigma_\varepsilon^2$

… in "in-sample error" model …

# Trading off Bias for Variance

- What is the best estimator for the given linear additive model?
- Least squares estimator
  $\ell(x_0) = x_0^T \underline{\beta} \quad \underline{\beta} = (X^T X)^{-1} X^T \mathbf{y}$
  is BLUE: Best Linear Unbiased Estimator
  - Optimal variance, wrt unbiased estimators
  - But variance is $O(\, p \,/\, N \,)$ …
- So if FEWER features, smaller variance…
  … albeit with some bias??

# Feature Selection

- LS solution can have large variance
  - variance $\propto$ p (#features)
- Decrease p $\Rightarrow$ decrease variance… but increase bias
- If decreases test error, do it!

  $\Rightarrow$ Feature selection
- Small #features also means:
  - easy to interpret

# Statistical Significance Test

- $\underline{Y} = \beta_0 + \sum_j \beta_j X_j$
- Q: Which $X_j$ are relevant?

  A: Use statistical hypothesis testing!
- Use simple model:

  $Y = \beta_0 + \sum_j \beta_j X_j + \varepsilon \qquad \varepsilon \sim N(0, \sigma_e^2)$
- Here: $\hat{\beta} \sim N(\beta, (X^T X)^{-1} \sigma_e^2)$
- Use $$z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}}$$

$$\hat{\sigma} = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

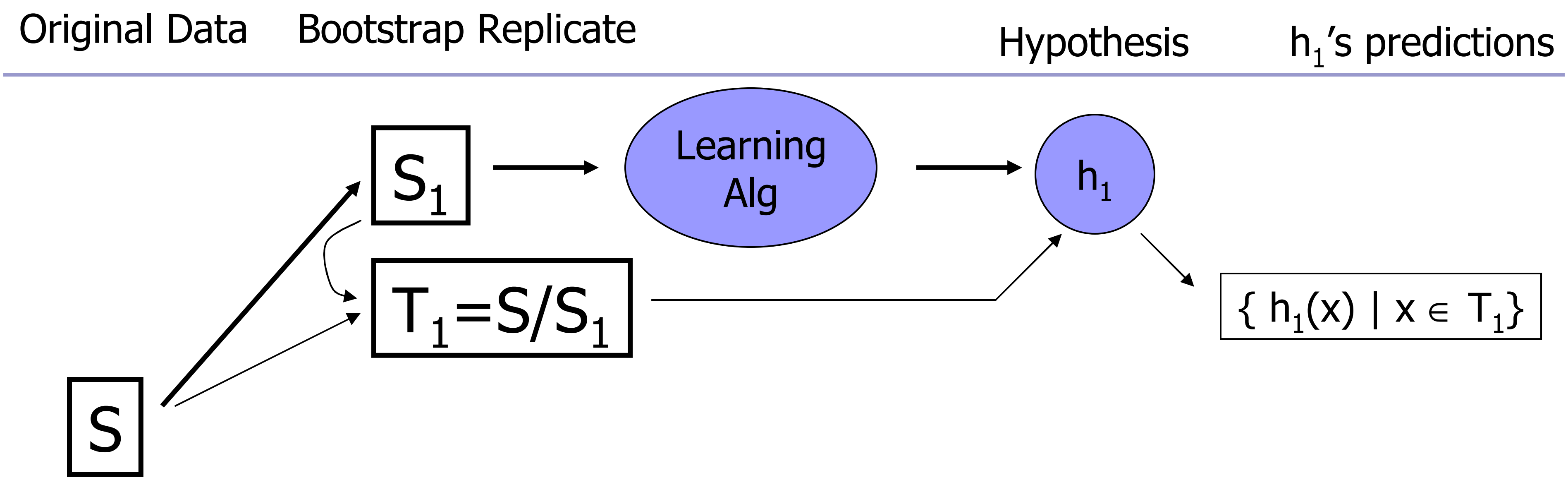
$v_j$ is the $j^{th}$ diagonal element of $(\mathbf{X}^T\mathbf{X})^{-1}$

- Keep variable $X_i$ if $z_j$ is large…

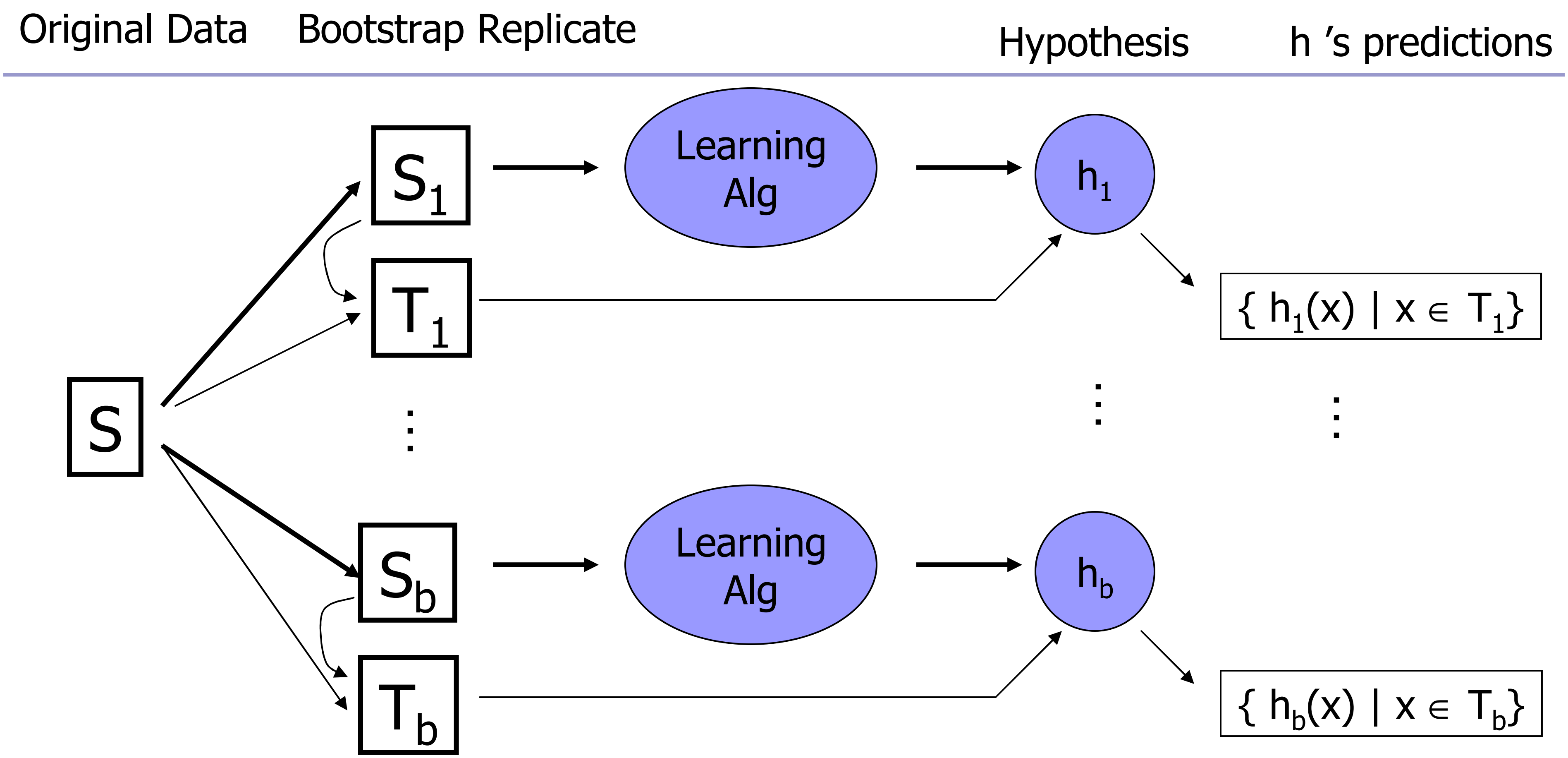
# Measuring Bias and Variance

- In practice (unlike in theory), only *ONE training set D*
  - Simulate multiple training sets by **bootstrap replicates**
    - D' = {x | x is drawn at random with replacement from D }
    - |D'| = |D|

# Estimating Bias / Variance



# Estimating Bias / Variance




- Each $S_i$ is bootstrap replicate
- $T_i = S / S_i$
- $h_i$ = hypothesis, based on $S_i$

# Average Response for each $x_i$

| | $x_1$ | … | $x_r$ |
|---|---|---|---|
| $\in^? T_1$ | $h_1(x_1)$ | … | |
| $\in^? T_2$ | -- | … | $h_2(x_r)$ |
| ⋮ | | | |
| $\in^? T_b$ | $h_b(x_1)$ | … | $h_b(x_r)$ |

$\underline{h(x_1)} = 1/k_1 \; \Sigma \; h_i(x_1)$ … $\underline{h(x_r)} = 1/k_r \; \Sigma \; h_i(x_r)$

$$\underline{h(x_j)} = \Sigma_{\{i:\, x \in Ti\}} h_i(x_j) \,/\, ||\{i:\, x \in T_i\}||$$

# Procedure for Measuring Bias and Variance

- Construct B bootstrap replicates of S

  $S_1, \ldots, S_B$
- Apply learning alg to each replicate $S_b$ to obtain hypothesis $h_b$
- Let $T_b = S \setminus S_b$ = data points not in $S_b$ (*out of bag* points)
- Compute predicted value

  $h_b(x)$

  for each $x \in T_b$

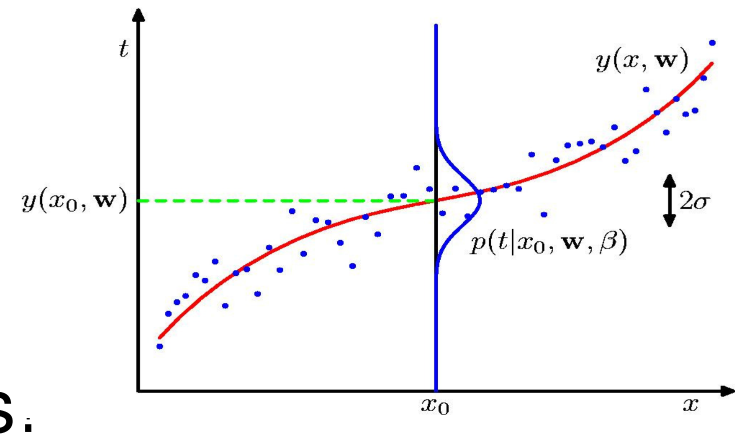
# Estimating Bias and Variance

- For each $x \in S$,
  - observed response $y$
  - predictions $y_1, \ldots, y_k$
- Compute average prediction $\underline{h(x)}$ $= ave_i \{y_i\}$
- Estimate bias: $\underline{h(x)} - y$
- Estimate variance:

$$\Sigma_{\{i:\, x \in Ti\}} \left( h_i(x) - \underline{h(x)} \right)^2 / (k-1)$$

- Assume noise is 0

# Outline




- Linear Regression
  - MLE = Least Squares!
  - Basis functions
- Evaluating Predictors
  - Training set error vs Test set error
  - Cross Validation
- Model Selection
  - Bias-Variance analysis
  - Regularization, Bayesian Model

# Regularization

- **Idea**: Penalize overly-complicated answers
- Regular regression minimizes:

$$\sum_i \left( y(\mathbf{x}^{(i)};\mathbf{w}) - t_i \right)^2$$

- Regularized regression minimizes:

$$\sum_i \left( y(\mathbf{x}^{(i)};\mathbf{w}) - t_i \right)^2 + \lambda \|\mathbf{w}\|$$

Note: May exclude constants from the norm

# Regularization: Why?

- For polynomials,
extreme curves typically require extreme values
- In general, encourages use of few features
  - only features that lead to a substantial increase in performance
- Problem: How to choose $\lambda$

# Solving Regularized Form

$$\textit{Solving} \quad w^* = \arg\min_w \left[ \sum_j \left[ t^j - \sum_i w_i x_i^j \right]^2 \right]$$

$$\boxed{\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}}$$

$$\textit{Solving} \quad w^* = \arg\min_w \left[ \sum_j \left[ t^j - \sum_i w_i x_i^j \right]^2 + \lambda \sum_i w_i^2 \right]$$

$$\boxed{\mathbf{w}^* = (\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}\mathbf{X}^T\mathbf{t}}$$

# Regularization: Empirical Approach

- Problem:
  magic constant λ trading-off complexity vs. fit
- Solution 1:
  - Generate multiple models
  - Use lots of test data to discover and discard bad models
- Solution 2: k-fold cross validation:
  - Divide data S into *k* subsets { $S_1$, …, $S_k$ }
  - Create validation set $S_{-i} = S_i - S$
    - Produces k groups, each of size (k -1)/k
  - For i=1..k: Train on $S_{-i}$, Test on $S_i$
  - Combine results … mean? median? …

# A Bayesian Perspective

- Given a space of possible hypotheses $H=\{h_j\}$
- Which hypothesis has the highest posterior:

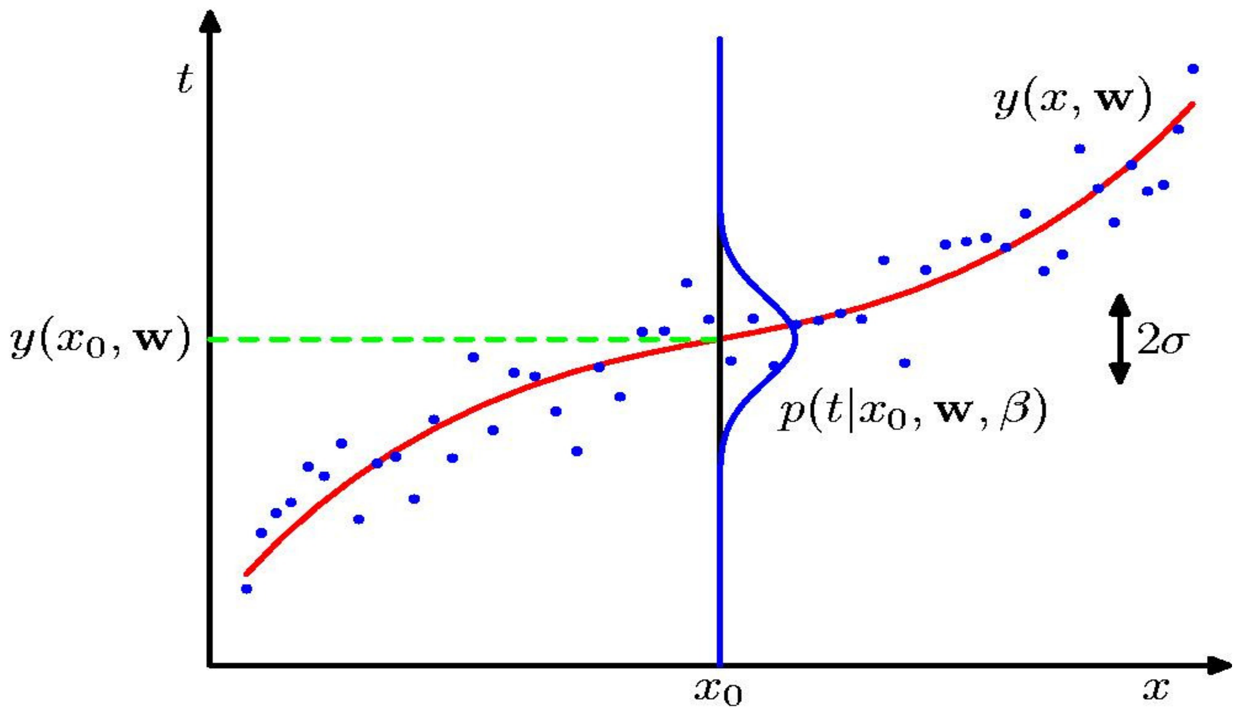
$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

- As P(D) does not depend on h:

  argmax P(h|D) = argmax P(D|h) P(h)
- "Uniform P(h)" ⇒ Maximum Likelihood Estimate
  - (model for which data has highest prob.)
- … can use P(h) for regularization …

# Bayesian Regression

- Assume that, given **x**, noise is iid Gaussian
- Homoscedastic noise model
  (same σ for each position)



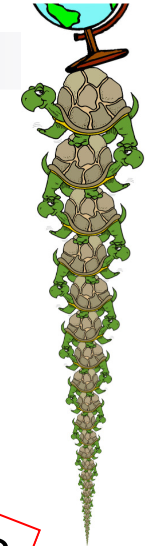
# Maximum Likelihood Solution

$$P(D \mid h) = P(t^{(1)},\ldots,t^{(m)} \mid y(\mathbf{x};\mathbf{w}),\sigma) \quad = \quad \prod_i \frac{e^{\frac{-(t^{(i)}-y(\mathbf{x};\mathbf{w}))^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}$$

MLE fit for mean is

- just linear regression fit
- does not depend upon $\sigma^2$

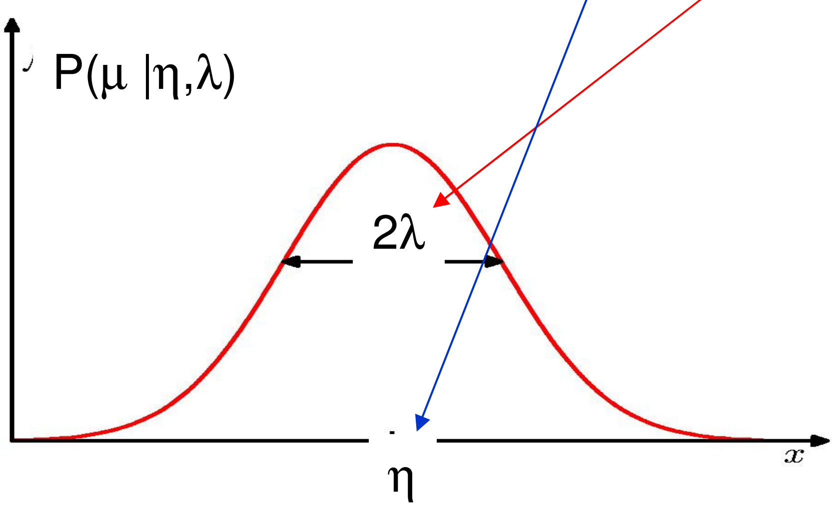
# Bayesian learning of Gaussian parameters



- Conjugate priors
  - Mean: Gaussian prior
  - Variance: Wishart Distribution
- Prior for mean:

Remember this??

$$P(\mu \mid \eta, \lambda) = \frac{1}{\lambda\sqrt{2\pi}} e^{\frac{-(\mu-\eta)^2}{2\lambda^2}}$$



# Bayesian Solution

- Introduce prior distribution over weights

$$p(h) = p(\mathbf{w} \mid \lambda) = N\left(\mathbf{w} \mid 0, \lambda^2 I\right)$$

- Posterior now becomes:

$$P(D \mid h)P(h) = P(t^{(1)},\ldots,t^{(m)} \mid y(\mathbf{x};\mathbf{w}),\sigma)\; P(\mathbf{w})$$

$$= \prod_i \frac{e^{\frac{-(t^{(i)}-y(\mathbf{x}^{(i)};\mathbf{w}))^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \; \frac{e^{\frac{-w^T w}{2\lambda^2}}}{\sqrt{2\pi\lambda^2}^{\,k}}$$

# Regularized Regression vs Bayesian Regression

- Regularized Regression minimizes:

$$\sum_i \left(t_i - y(\mathbf{x}^{(i)};\mathbf{w})\right)^2 + \kappa\|\mathbf{w}\|$$

- Bayesian Regression maximizes:

$$\boxed{const + \sum_i \frac{-(t^{(i)} - y(\mathbf{x}^{(i)};\mathbf{w}))^2}{2\sigma^2} + \frac{-\mathbf{w}^T\mathbf{w}}{2\lambda^2}}$$

- These are identical (up to constants)

… take log of Bayesian regression criterion

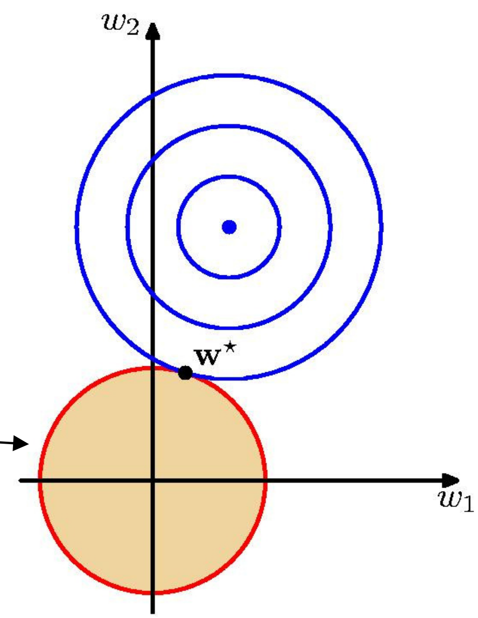
# Viewing $L_2$ Regularization

$$w^* = \arg\min_w \left[ \sum_j \left[ t^j - \sum_i w_i x_i^j \right]^2 + \lambda \sum_i w_i^{\,2} \right]$$

- Using Lagrange Multiplier…

$$\Rightarrow \quad w^* = \arg\min_w \left[ \sum_j \left[ t^j - \sum_i w_i x_i^j \right]^2 \right]$$

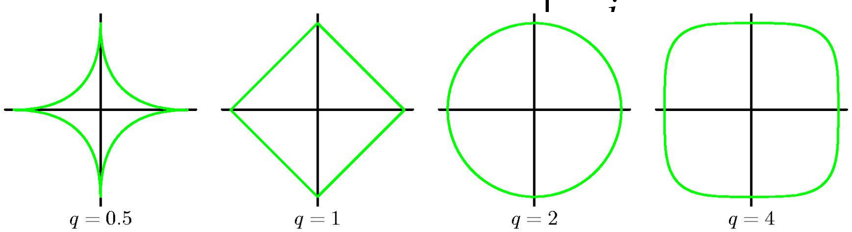
s.t. $\quad \sum_i w_i^{\,2} \le \omega$

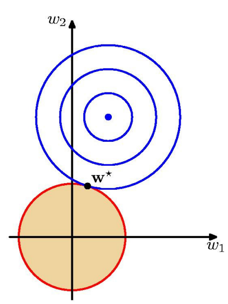


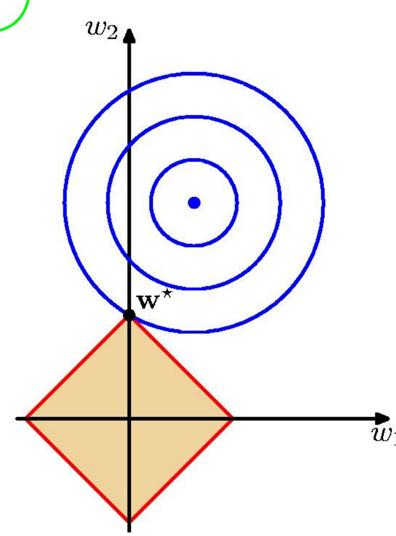
# Use $L_2$ vs $L_1$ Regularization

$$w^* = \arg\min_w \left[ \sum_j \left[ t^j - \sum_i w_i x_i^j \right]^2 + \lambda \sum_i | w_i |^q \right]$$

$$\Rightarrow \quad w^* = \arg\min_w \left[ \sum_j \left[ t^j - \sum_i w_i x_i^j \right]^2 \right] \quad \text{s.t.} \quad \boxed{\sum_i | w_i |^q \leq \omega}$$




Intersections often on axis!
… so $w_i = 0$ !!

LASSO!

# What you need to know

- Regression
  - Optimizing sum squared error == MLE !
  - Basis functions = features
  - Relationship between regression and Gaussians
- Evaluating Predictor
  - TestSetError ≠ Prediction Error
  - Cross Validation
- Bias-Variance trade-off
  - Model complexity …

Play with Applet

- Regularization ≈ Bayesian modeling
- $L_1$ regularization – prefers 0 weights!