Assessing Performance

CSE 446: Machine Learning
Slides created by Emily Fox (mostly)

April 8, 2019

©2017 Emily Fox

To fit these more general functions

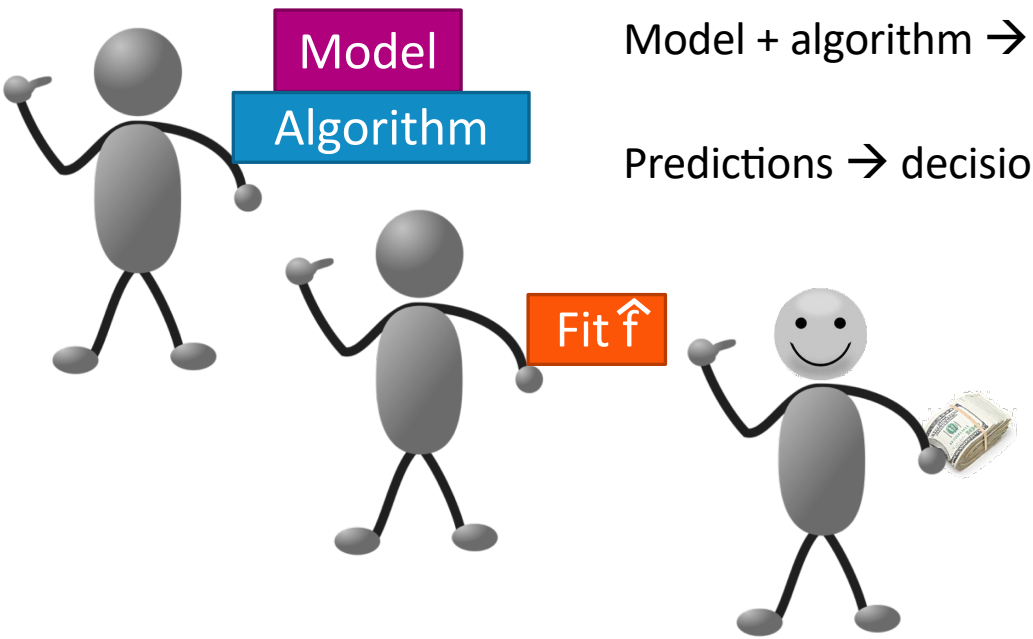
- Start with input features $\mathbf{x} = (x[1], x[2], \ldots, x[d])$

and training set: $\{(\mathbf{x}_i, y_i)\}_{i=1..n}$

- Define feature map that transforms each $\mathbf{x}_i$ to higher dimensional feature vector $\mathbf{h}(\mathbf{x}_i)$.
- Model: $y_i = \sum_{j=1}^{p} w_j \, h_j(\mathbf{x}_i) + \varepsilon_i$
- Find $\hat{\mathbf{w}}$ that minimizes RSS $= \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} w_j \, h_j(\mathbf{x}_i))^2$

  $= (\mathbf{y} - H\mathbf{w})^T(\mathbf{y} - H\mathbf{w})$
- Solution: $H^T H\hat{\mathbf{w}} = H^T y$

How good are my predictions?

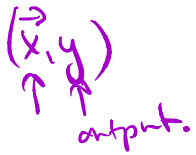


Model + algorithm → fitted function

Predictions → decisions → outcome

©2017 Emily Fox

Measuring loss



Loss function:

$L(y, f_{\hat{w}}(x))$

Cost of using $\hat{w}$ at x when y is true

actual value

$\hat{f}(x)$ = predicted value $\hat{y}$

Examples:

Squared error: $L(y, f_{\hat{w}}(x)) = (y - f_{\hat{w}}(x))^2$

Absolute error: $L(y, f_{\hat{w}}(x)) = |y - f_{\hat{w}}(x)|$

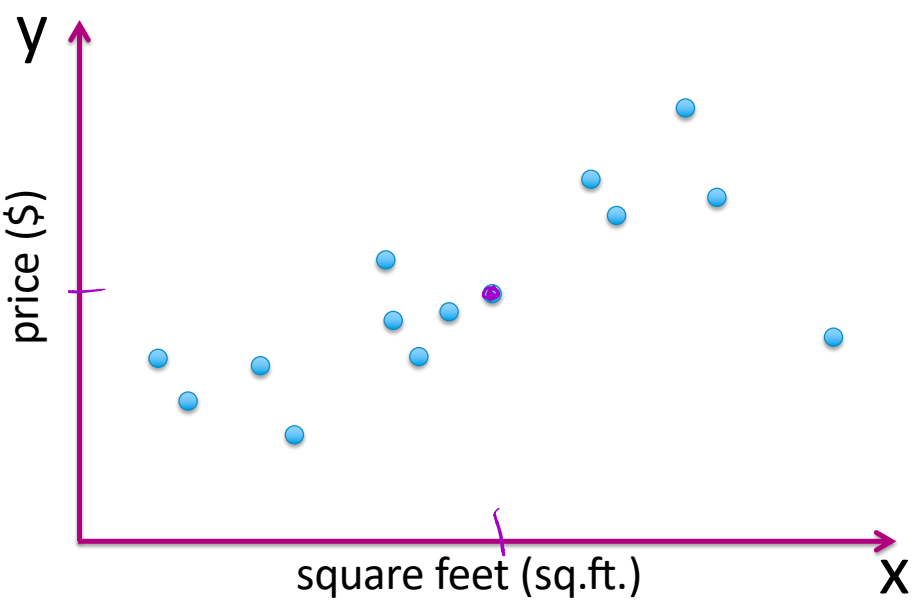
©2017 Emily Fox

"Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful." George Box, 1987.

Assessing the loss

Assessing the loss
Part 1: Training error

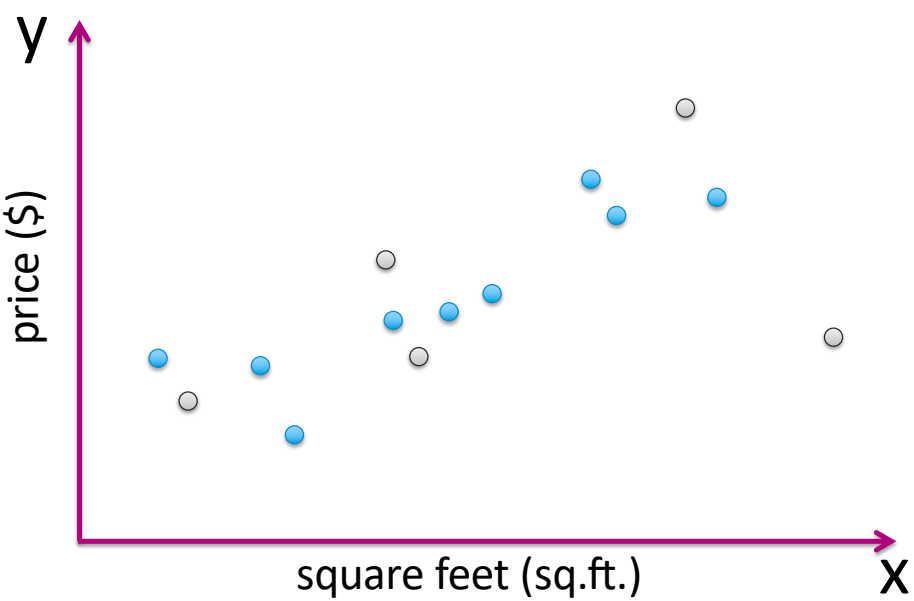
Start with a data set



©2017 Emily Fox

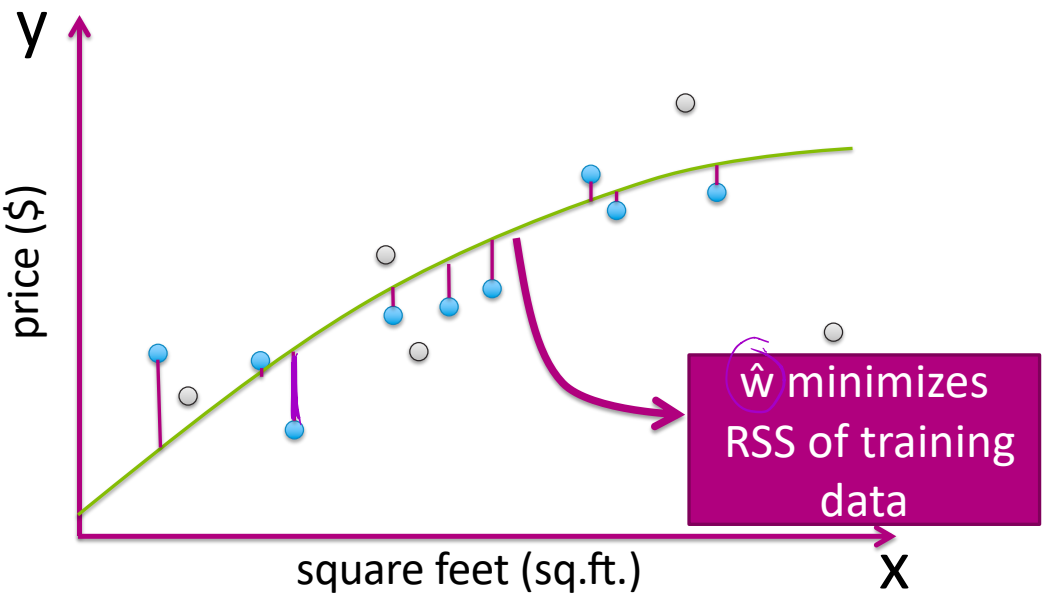
Define training data

$\{(\vec{x}_i, y_i)\}_{i=1}^{n}$



©2017 Emily Fox

Example:
Fit quadratic to minimize RSS



©2017 Emily Fox

Compute training error

1. Define a loss function $L(y, f_{\hat{w}}(x))$
   - E.g., squared error,…

2. Training error using $f_{\hat{w}}$

   = avg. loss on houses in training set

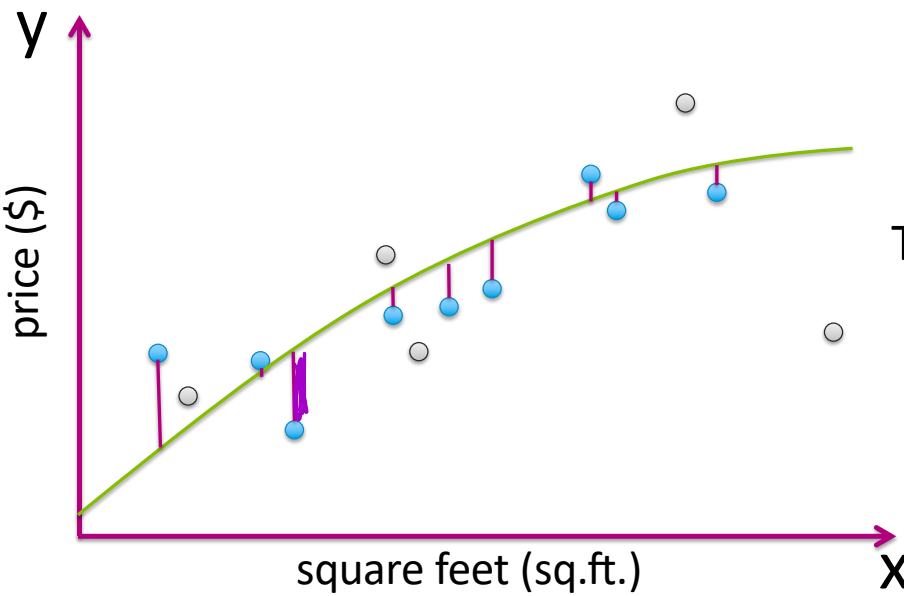
   = $\frac{1}{n}\sum_{i=1}^{n} L(y_i, f_{\hat{w}}(x_i))$

fit using training data

©2017 Emily Fox

Example:

Use squared error loss $(y-f_{\hat{w}}(x))^2$



Training error ($\hat{w}$) = 1/n *

$[(\$_{train\ 1}-f_{\hat{w}}(sq.ft._{train\ 1}))^2$

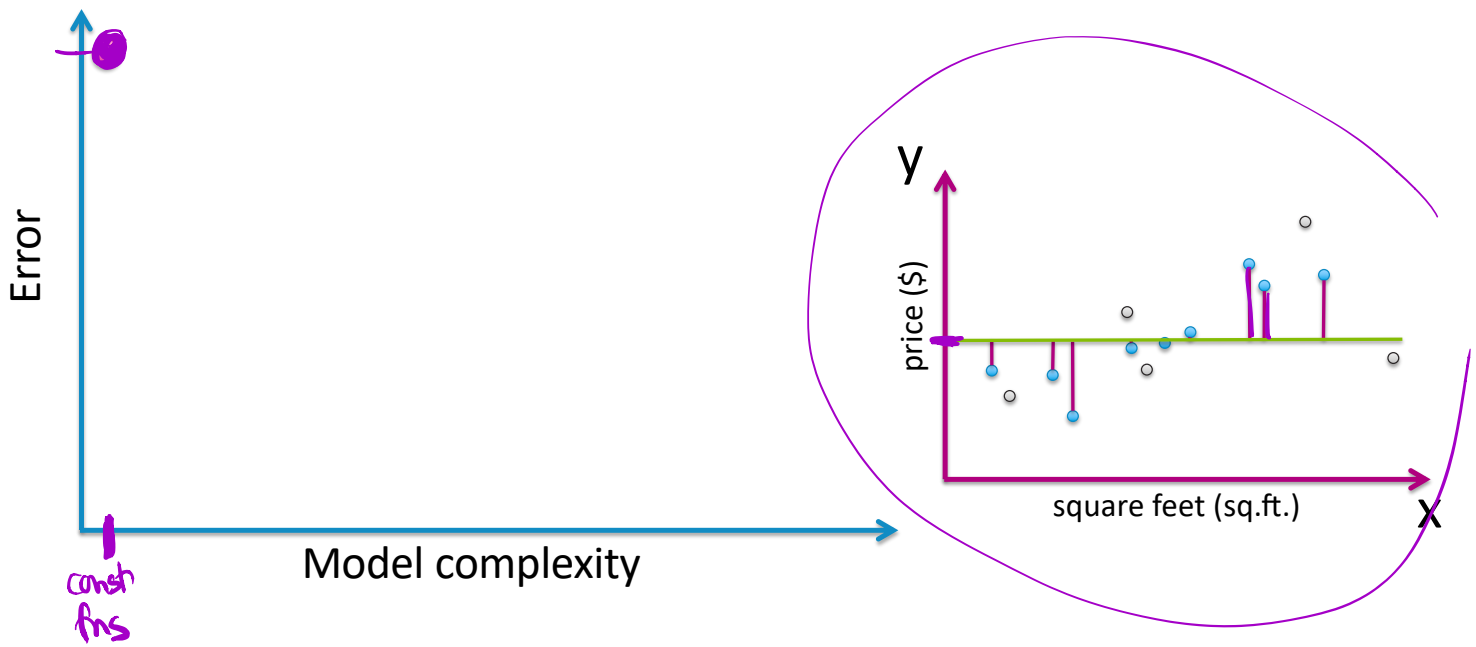
$+ (\$_{train\ 2}-f_{\hat{w}}(sq.ft._{train\ 2}))^2$

$+ (\$_{train\ 3}-f_{\hat{w}}(sq.ft._{train\ 3}))^2$

+ … include all training houses]

©2017 Emily Fox

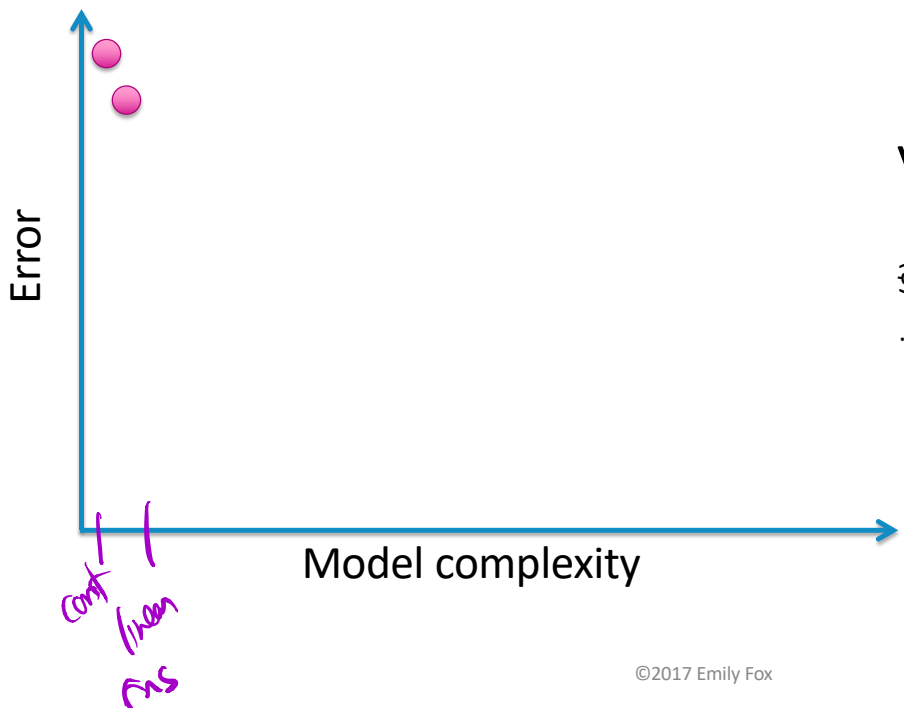
Training error vs. model complexity

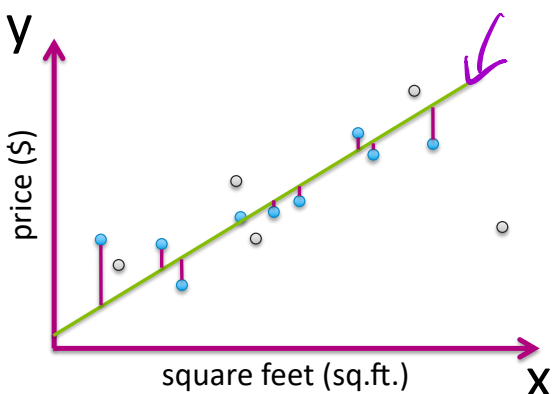


©2017 Emily Fox

Training error vs. model complexity

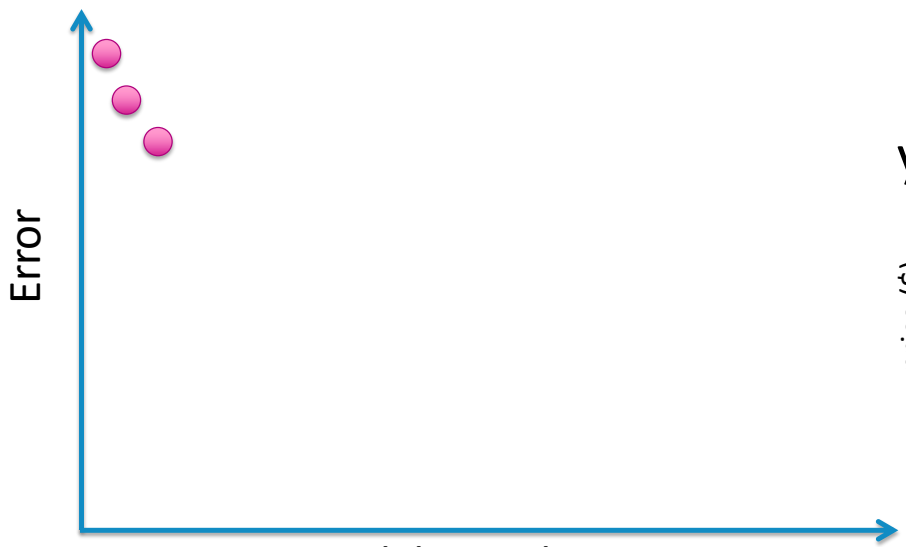
blue pts - training set.

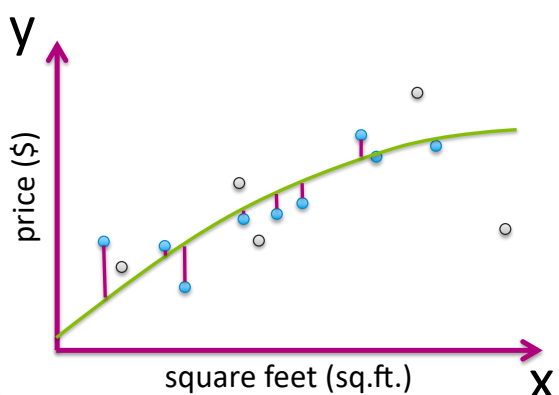




©2017 Emily Fox

Training error vs. model complexity

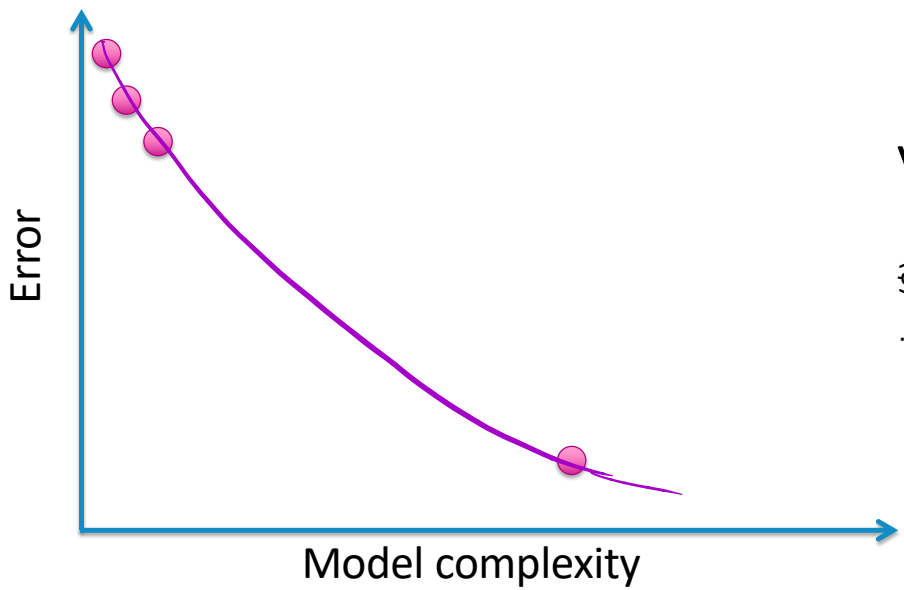


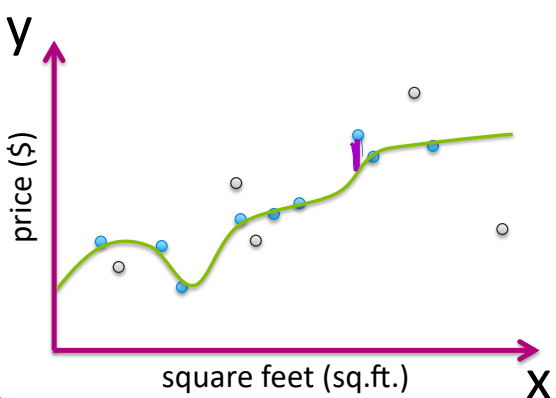


polynomials of higher & higher degree

©2017 Emily Fox

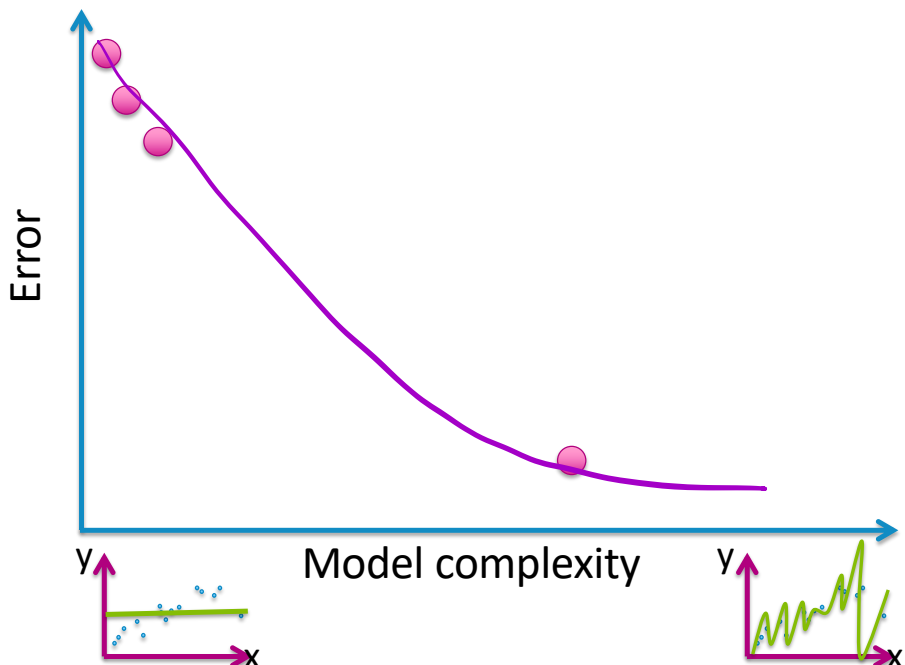
Training error vs. model complexity





©2017 Emily Fox

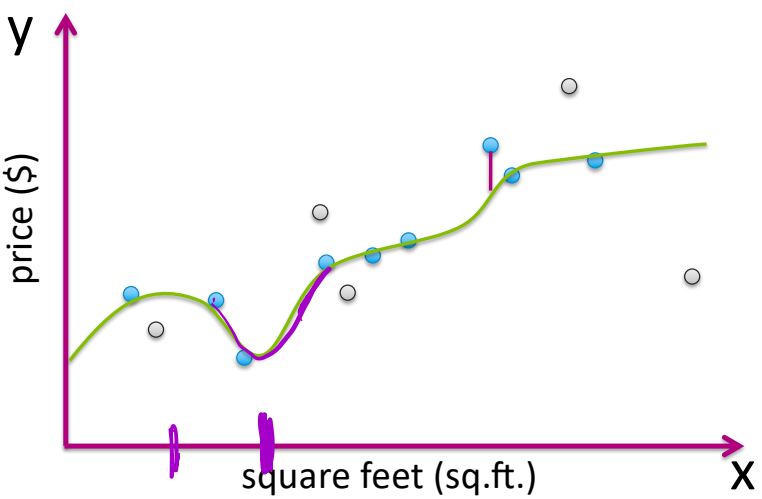
Training error vs. model complexity



©2017 Emily Fox

Is training error a good measure of predictive performance?

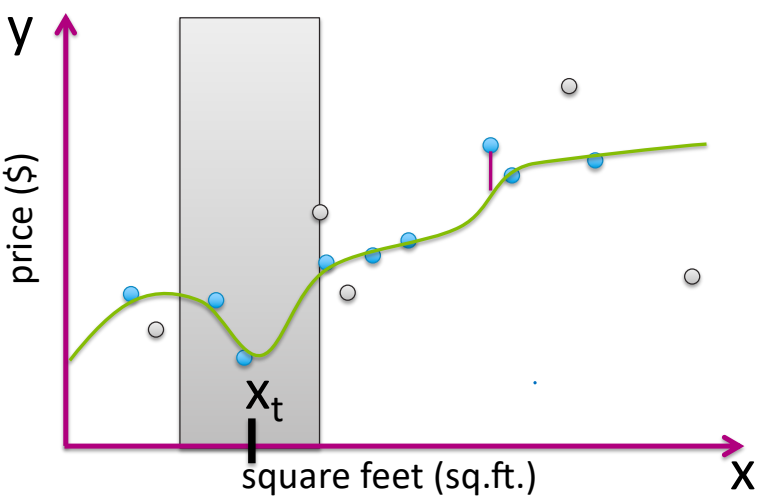
How do we expect to perform on a new house?



©2017 Emily Fox

Is training error a good measure of predictive performance?

Is there something particularly bad about having x_t sq.ft.??

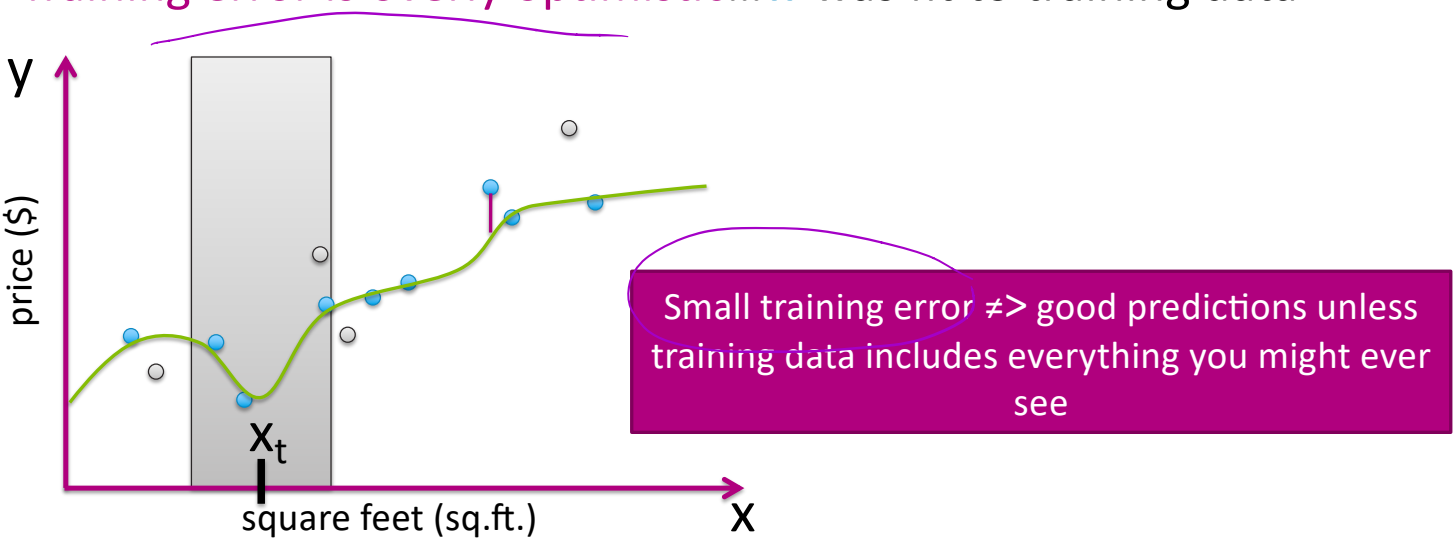


©2017 Emily Fox

Is training error a good measure of predictive performance?

Issue:

Training error is overly optimistic…$\hat{w}$ was fit to training data

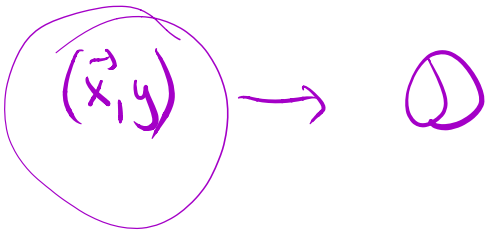


Small training error ≠> good predictions unless training data includes everything you might ever see

©2017 Emily Fox

Assessing the loss
Part 2: Generalization (true) error

Generalization error



Really want estimate of loss over all possible (🏠,$) pairs

Lots of houses
in neighborhood,
but not in dataset

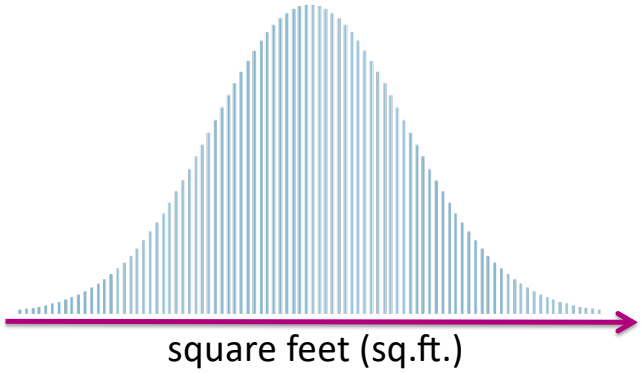
©2017 Emily Fox

$p(\vec{x}, y)$

Distribution over houses

$p(\vec{x})$

In our neighborhood, houses of what # sq.ft. ( ) are we likely to see?



©2017 Emily Fox

$(\vec{x}, y)$

$p(x,y) = p(x)\,p(y|x)$

$p(y|x)$

Distribution over sales prices

For houses with a given # sq.ft. ( ), what house prices $ are we likely to see?

©2017 Emily Fox

Generalization error definition

Really want estimate of loss over all possible (🏠,$) pairs

average over all possible (x,y) pairs weighted by how likely each is

Formally:

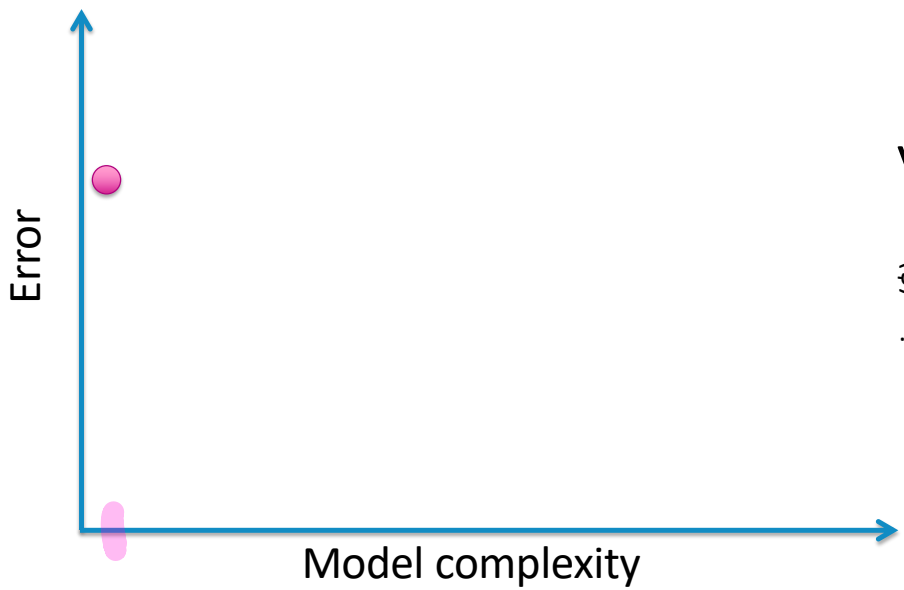
$$\text{generalization error} = E_{x,y}[L(y, f_{\hat{w}}(x))]$$

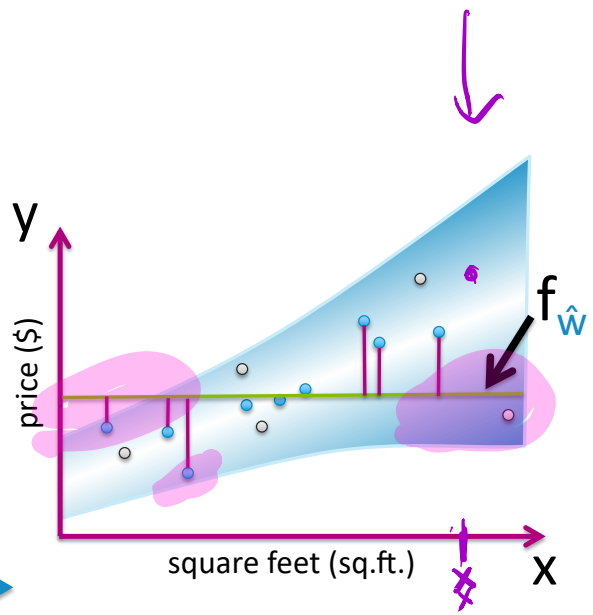
fit using training data

$$\int_{x,y} (y - f_{\hat{w}}(x))^2 p(x,y)\, dx\, dy$$

©2017 Emily Fox

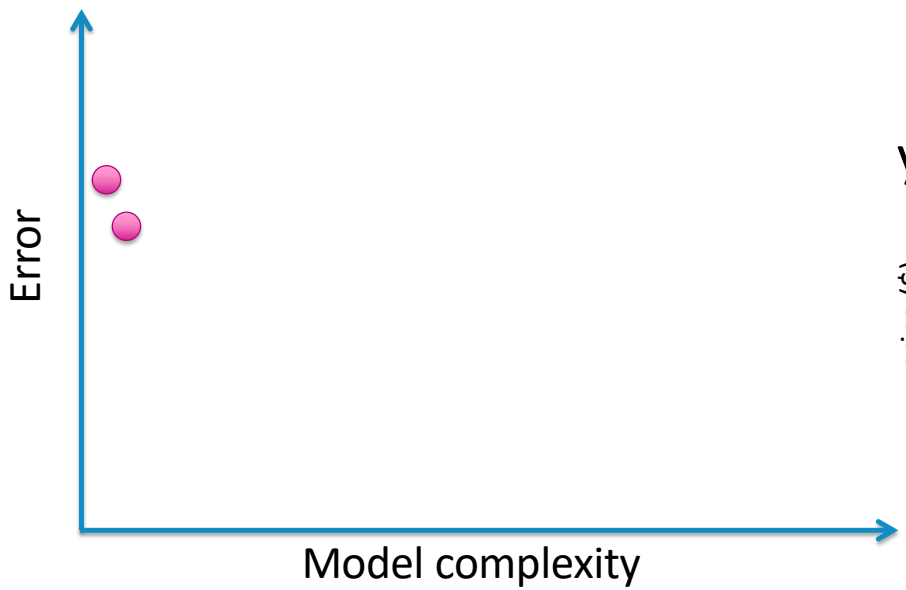
Generalization error vs. model complexity

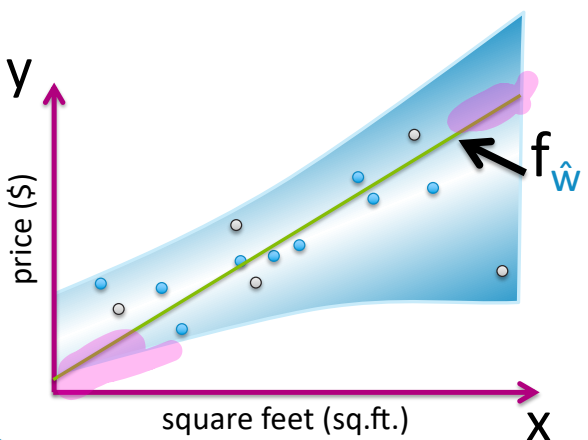




©2017 Emily Fox

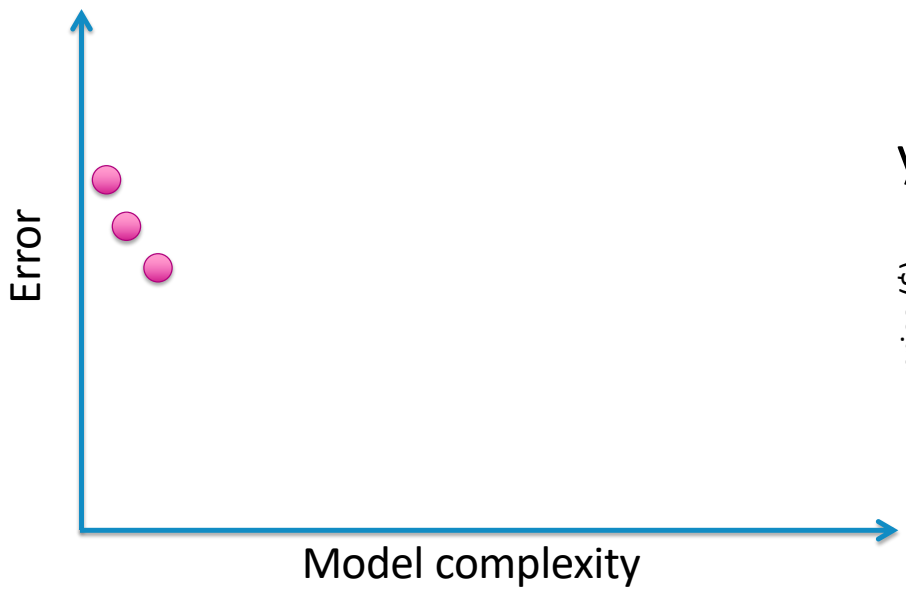
Generalization error vs. model complexity

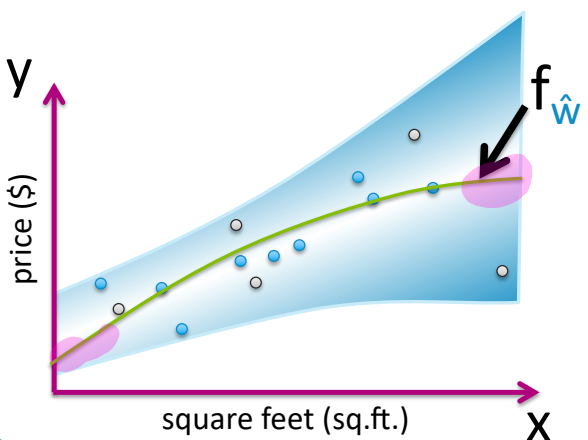




©2017 Emily Fox

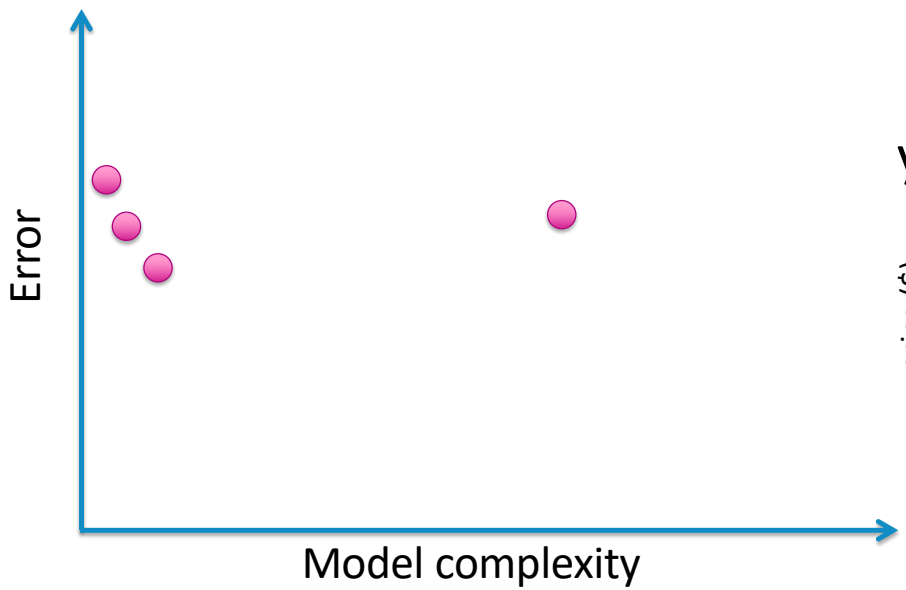
Generalization error vs. model complexity

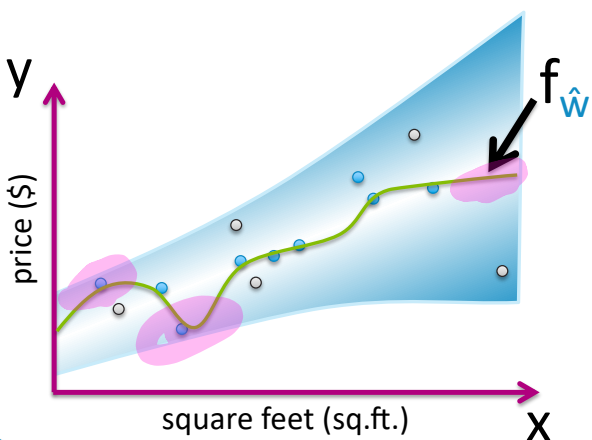




©2017 Emily Fox

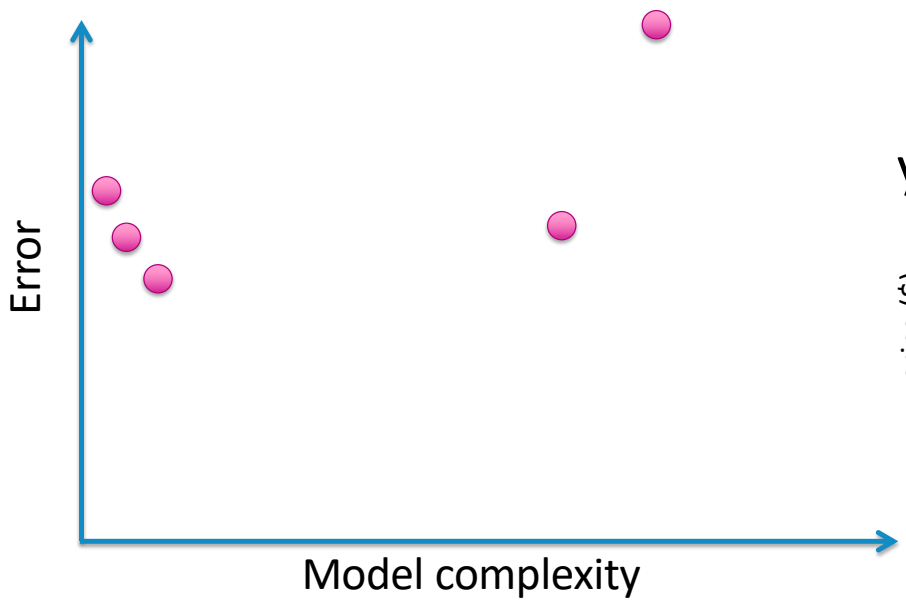
Generalization error vs. model complexity

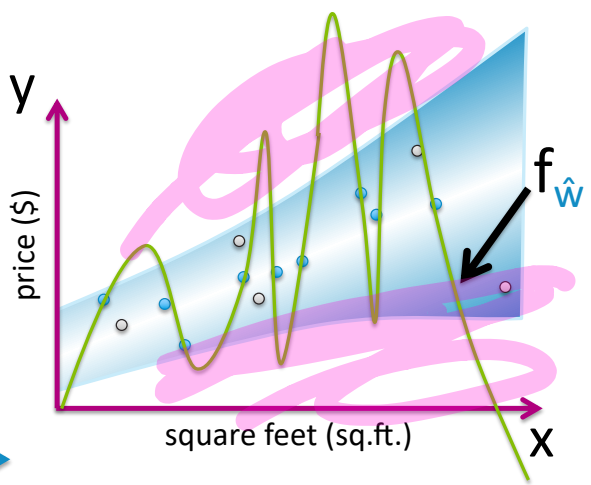




©2017 Emily Fox

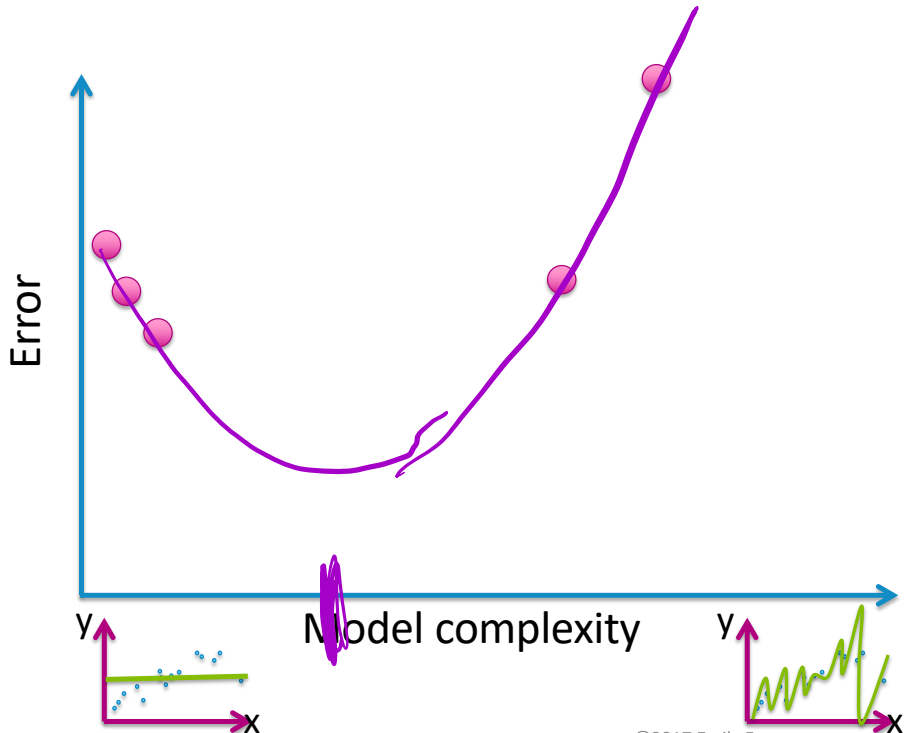
Generalization error vs. model complexity





©2017 Emily Fox

Generalization error vs. model complexity



Can't compute!

©2017 Emily Fox

Assessing the loss
Part 3: Test error

Approximating generalization error

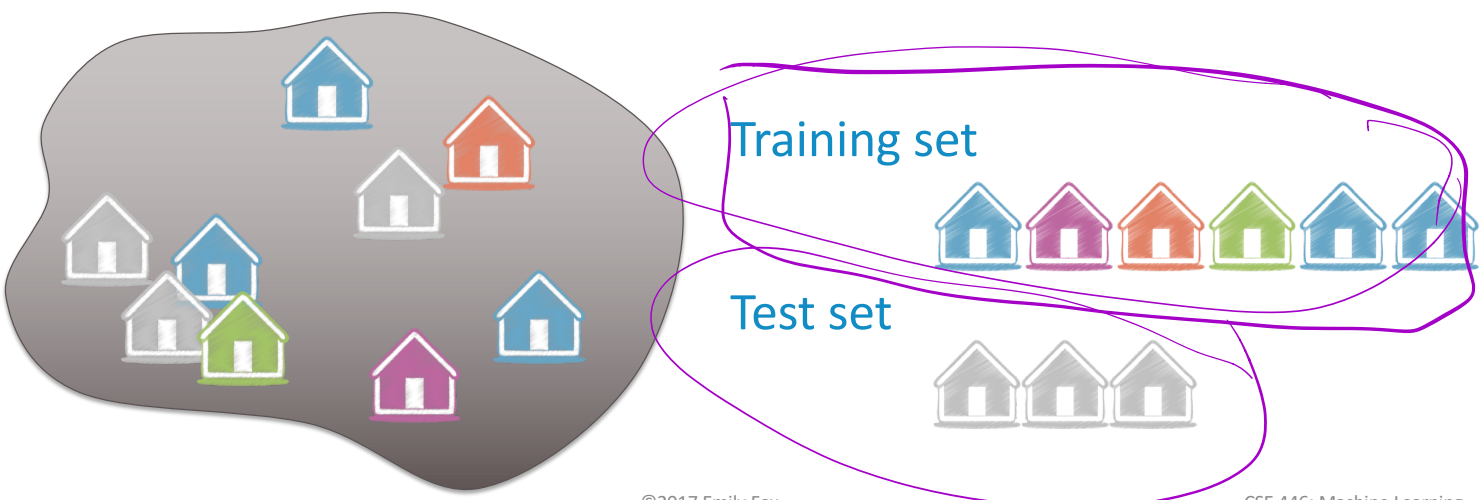
Wanted estimate of loss over all possible (🏠,$) pairs

Approximate by looking at houses not in training set

©2017 Emily Fox

Forming a test set

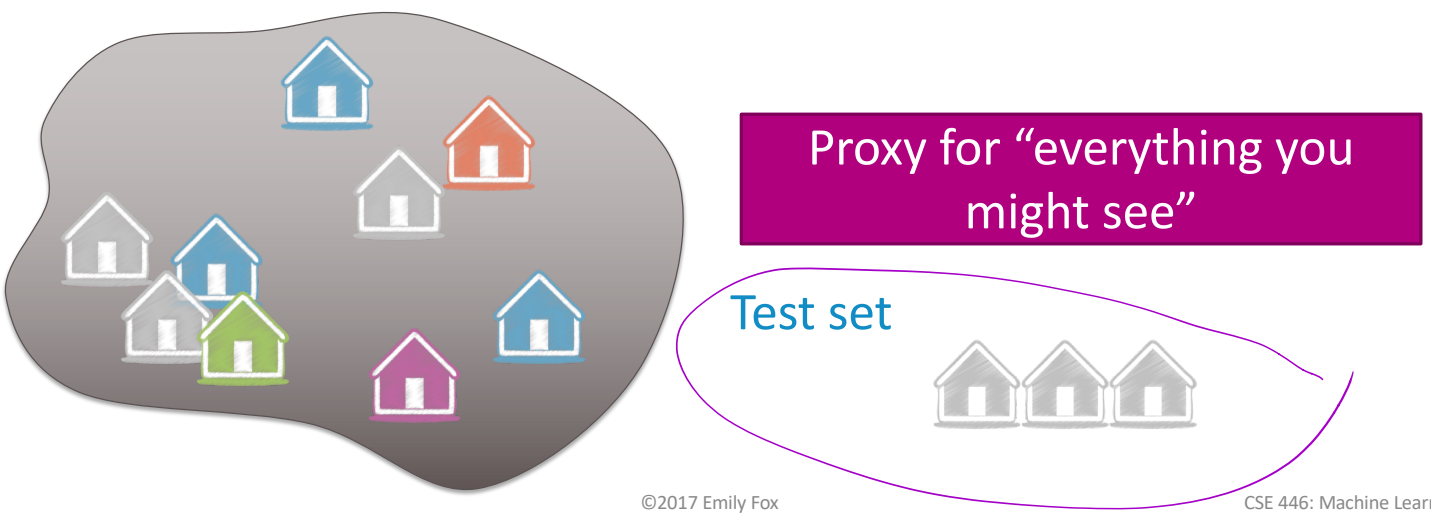
Hold out some (🏠,$) that are *not* used for fitting the model



©2017 Emily Fox

Forming a test set

Hold out some (🏠,$) that are *not* used for fitting the model



©2017 Emily Fox

every sample i.i.d.

Compute test error

Test error

= avg. loss on houses in test set

$$= \frac{1}{N_{test}} \sum_{i \text{ in test set}} L(y_i, f_{\hat{w}}(x_i))$$

test points

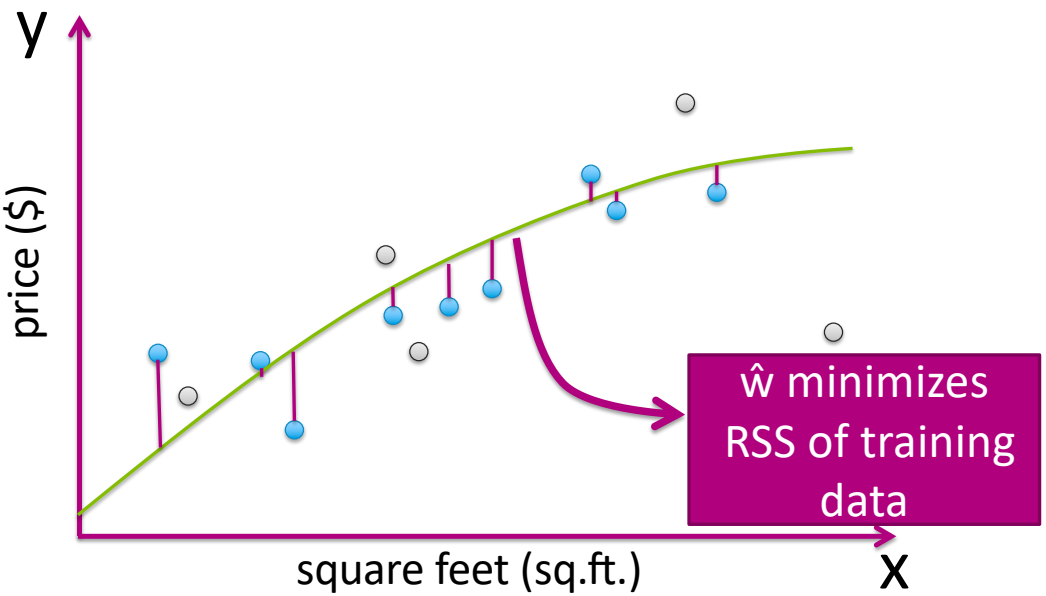
fit using training data

has never seen test data!

©2017 Emily Fox

Example:

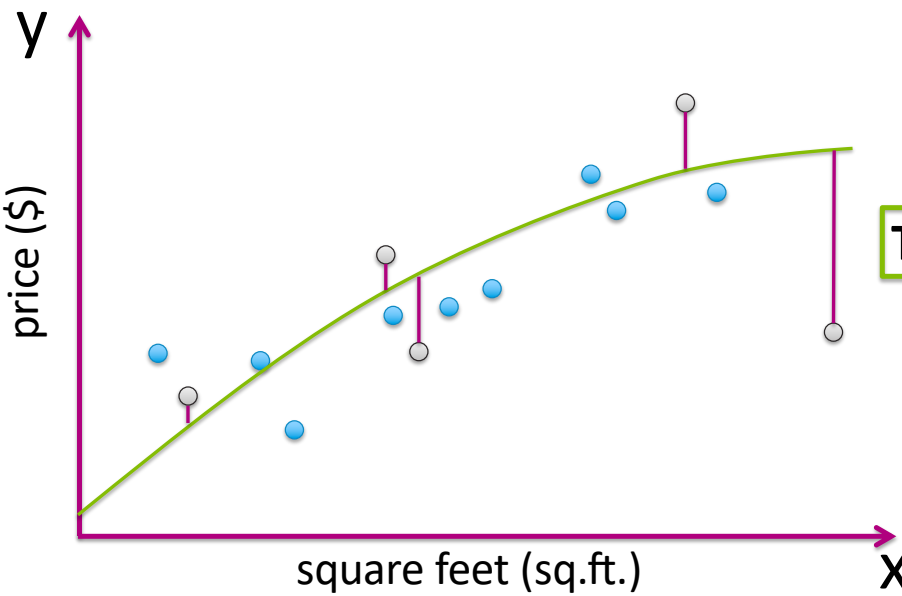
As before, fit quadratic to training data



©2017 Emily Fox

Assess performance using **test error**

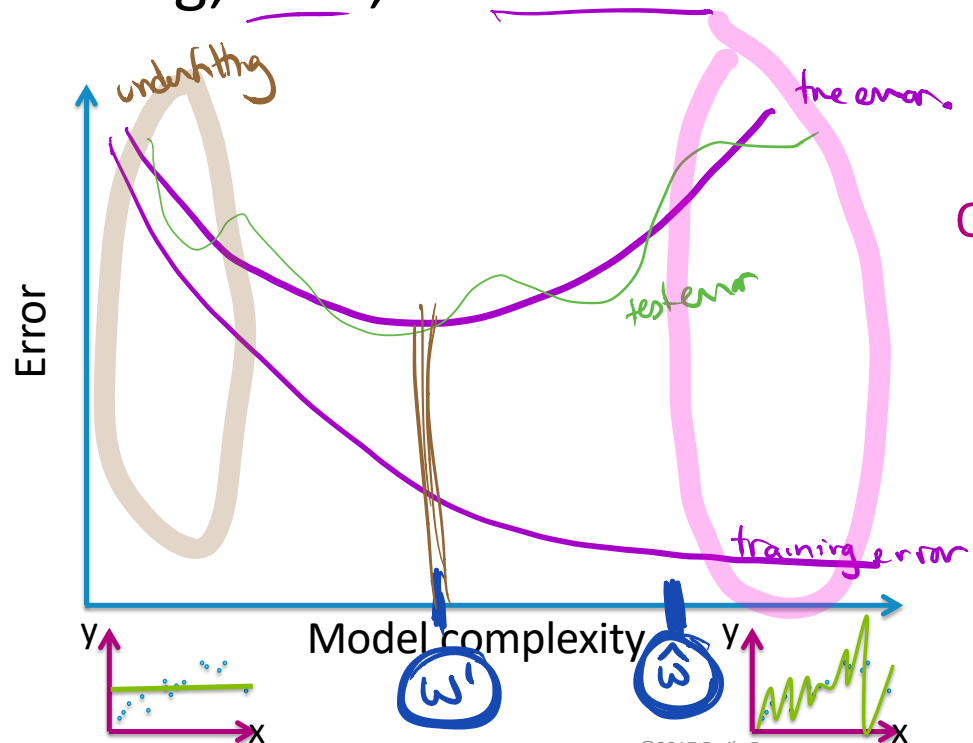
As before, use squared error loss $(y-f_{\hat{w}}(x))^2$



Test error ($\hat{w}$) = $1/N_{test}$ *
$[(\$_{test\ 1}-f_{\hat{w}}(sq.ft._{test\ 1}))^2$
$+ (\$_{test\ 2}-f_{\hat{w}}(sq.ft._{test\ 2}))^2$
$+ (\$_{test\ 3}-f_{\hat{w}}(sq.ft._{test\ 3}))^2$
+ … include all test houses]

©2017 Emily Fox

Training, true, & test error vs. model complexity



Overfitting if:

at $\hat{w}$

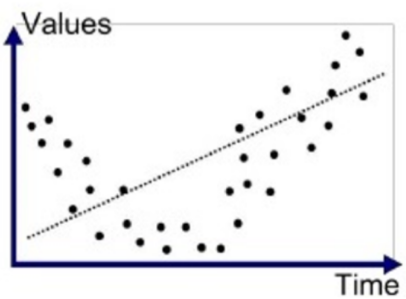
$\exists\, w'$ s.t.

$\text{trainerror}(\hat{w}) < \text{trainerror}(w')$

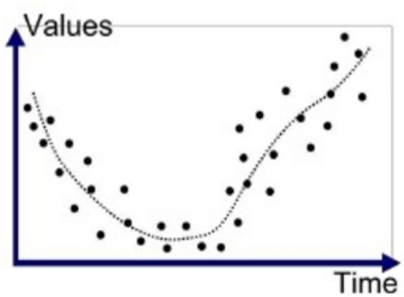
$\text{testerror}(\hat{w}) > \text{testerror}(w')$

©2017 Emily Fox

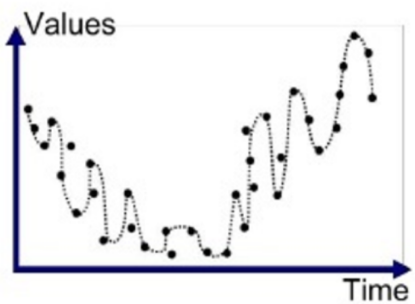
Underfitting vs Overfitting



Underfitted



Good Fit/Robust

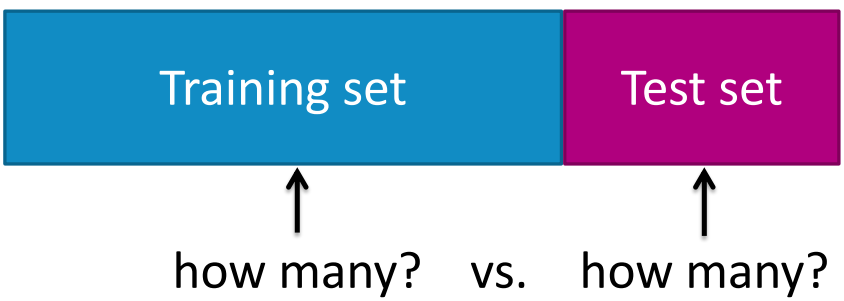


Overfitted

Courtesy Blog@AlgoTrading101

Training/test split

Training/test splits



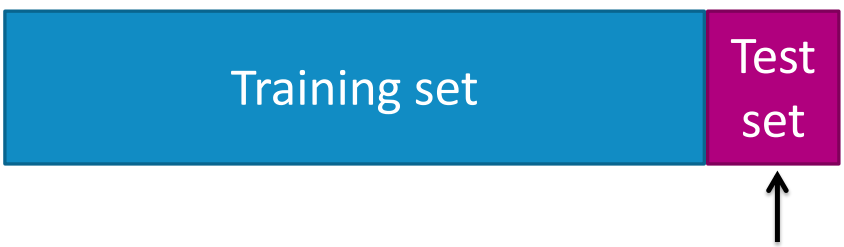
©2017 Emily Fox

Training/test splits

Too few → $\hat{w}$ poorly estimated

©2017 Emily Fox

Training/test splits



Too few → test error bad approximation of generalization error

©2017 Emily Fox

Training/test splits

Typically, just enough test points to form a reasonable estimate of generalization error

If this leaves too few for training, other methods like cross validation (will see later…)

©2017 Emily Fox

3 sources of error + the bias-variance tradeoff

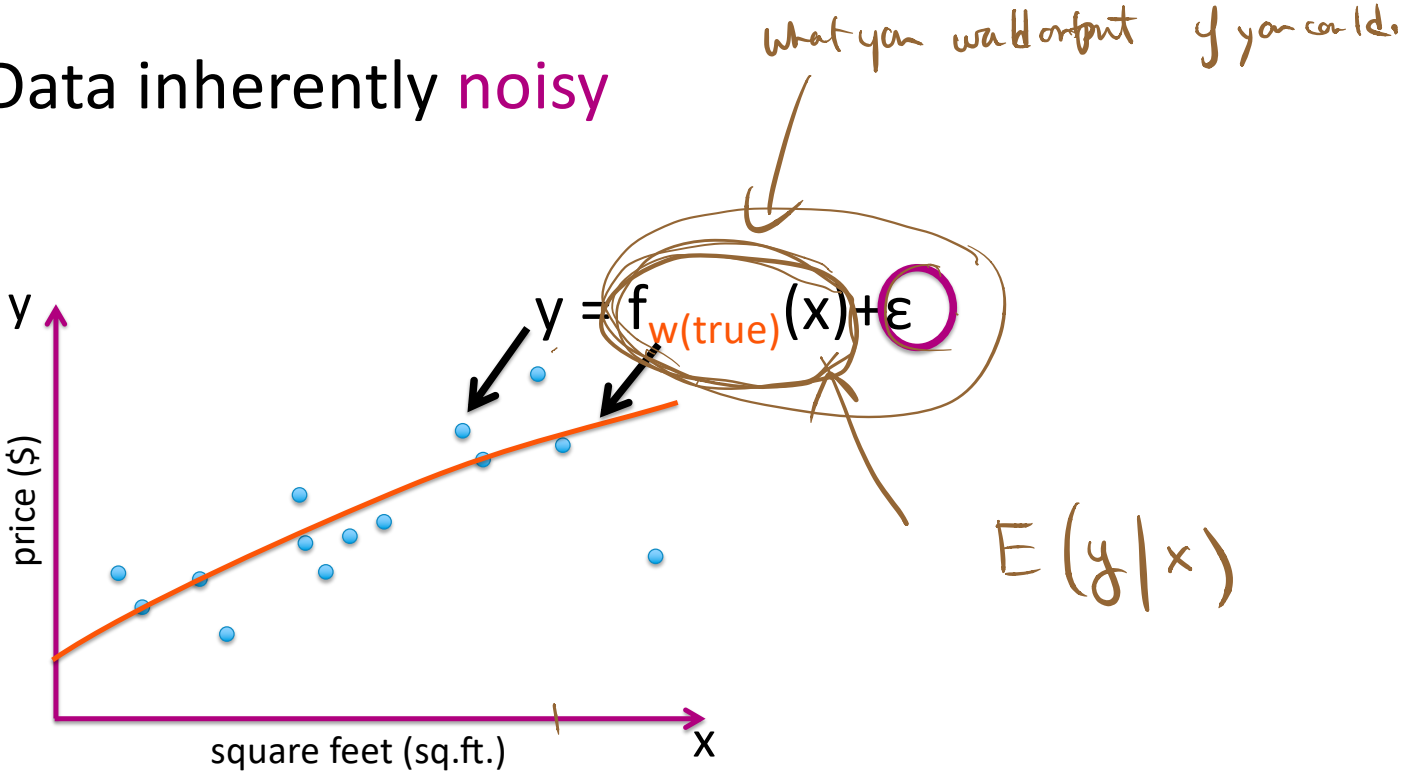
3 sources of error

In forming predictions, there are 3 sources of error:

1. Noise
2. Bias
3. Variance

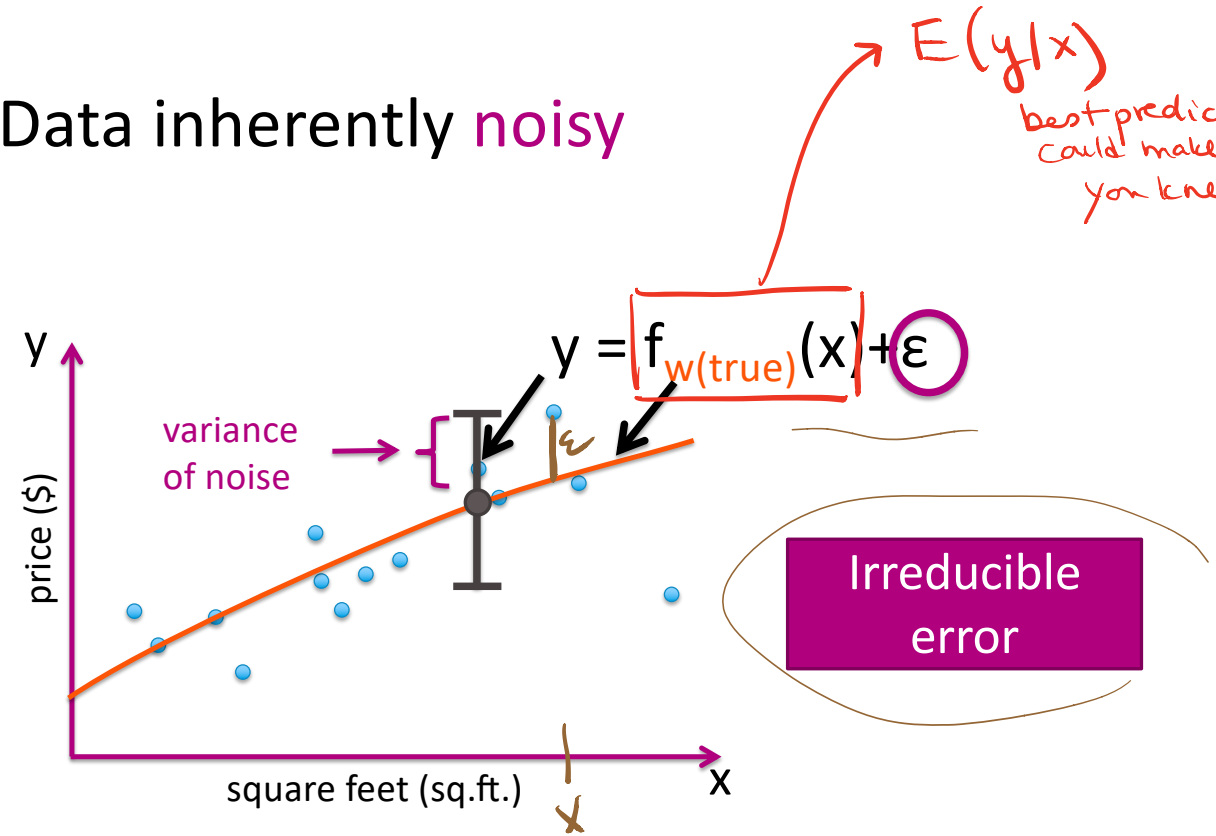
©2017 Emily Fox

Data inherently noisy



©2017 Emily Fox

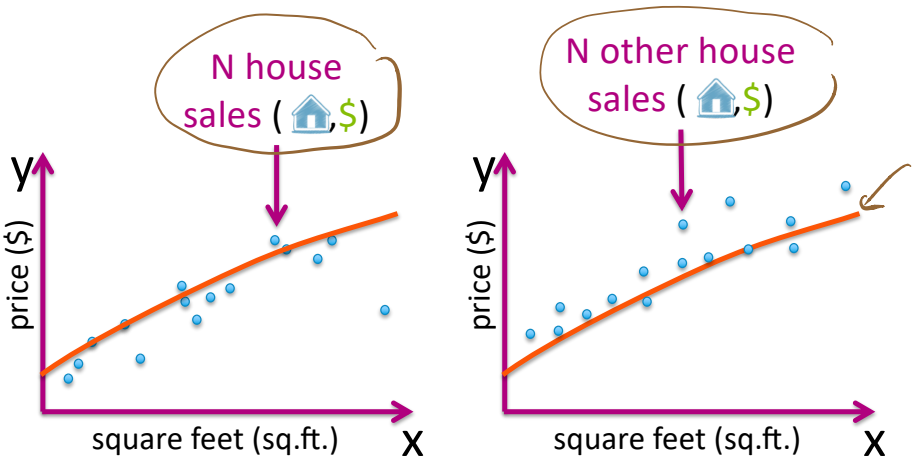
Data inherently noisy



©2017 Emily Fox

Bias contribution

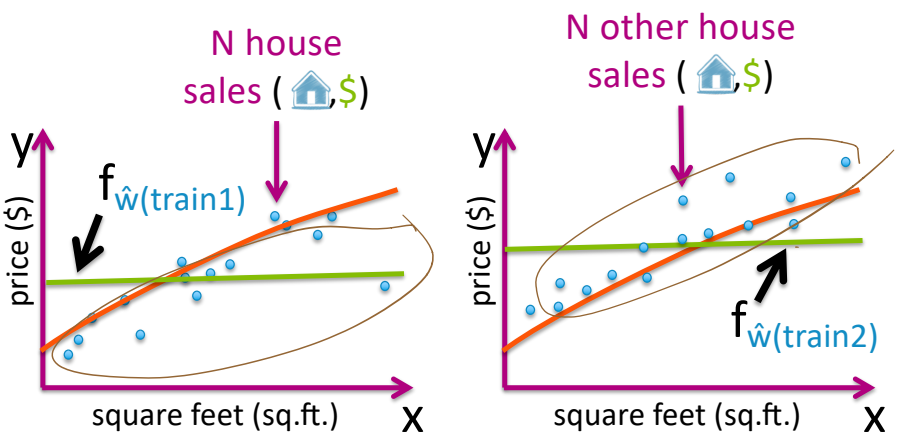
Assume we fit a constant function



©2017 Emily Fox

Bias contribution

Assume we fit a constant function

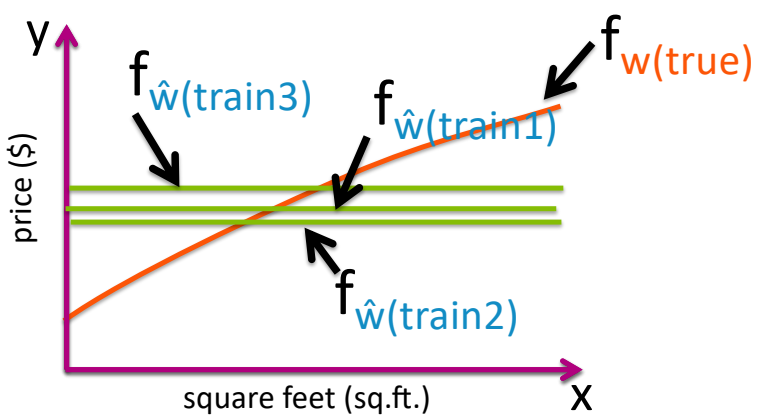


©2017 Emily Fox

Bias contribution

$$f_{\bar{w}}(x) = E_{train}\, f[_{\hat{w}(train)}(x)]$$

Over all possible size N training sets, what do I expect my fit to be?

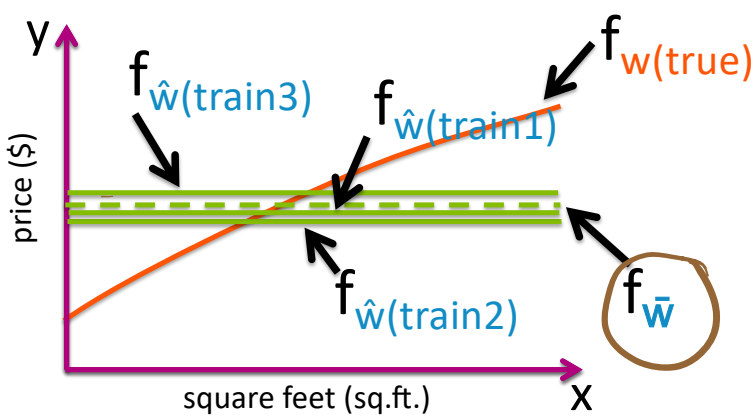


©2017 Emily Fox

Bias contribution

$f_{\bar{w}}(x) = E_{train}\, f[_{\hat{w}(train)}(x)]$

Over all possible size N training sets, what do I expect my fit to be?



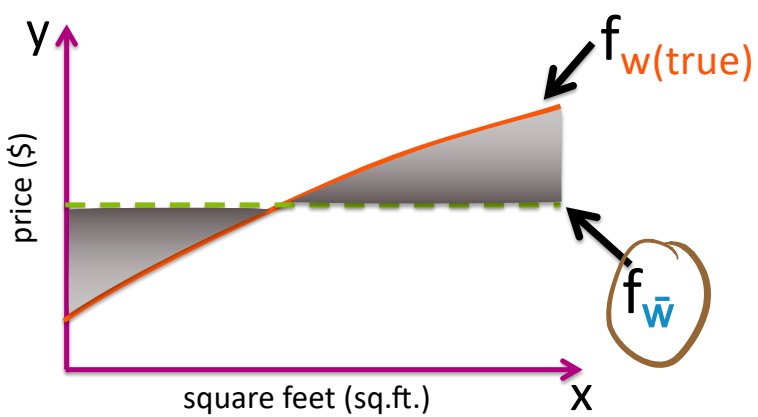
©2017 Emily Fox

Bias contribution

$$f_{\bar{w}}(x) = E_{train}\, f[\hat{w}_{(train)}(x)]$$

$$\text{Bias}(x) = f_{w(true)}(x) - f_{\bar{w}}(x)$$

Is our approach flexible enough to capture $f_{w(true)}$?
If not, error in predictions.



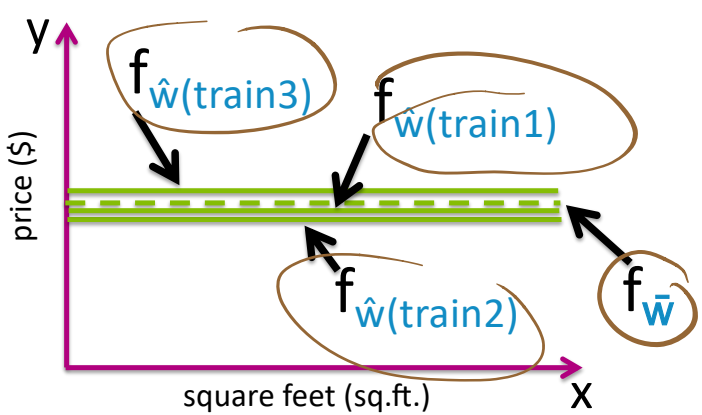
low complexity
→
high bias

©2017 Emily Fox

Variance contribution

$f_{\bar{w}}(x) = E_{train}\, f[_{\hat{w}(train)}(x)]$

How much do specific fits vary from the expected fit?

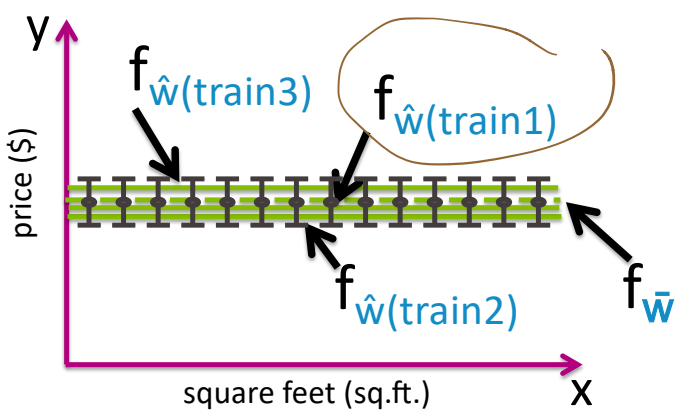


©2017 Emily Fox

Variance contribution

$$f_{\bar{w}}(x) = E_{train}\, f[_{\hat{w}(train)}(x)]$$

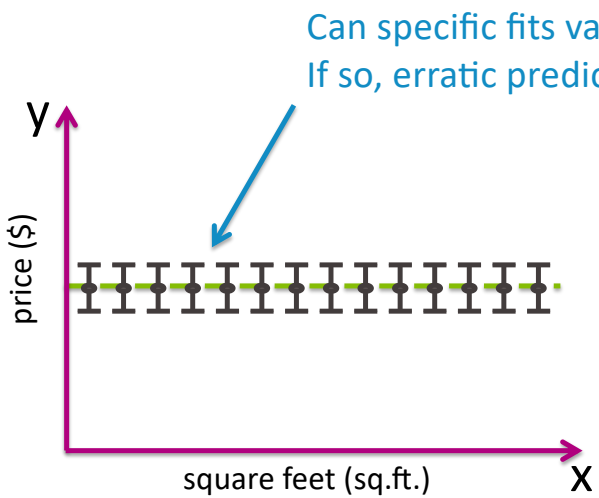
How much do specific fits vary from the expected fit?



©2017 Emily Fox

Variance contribution

How much do specific fits vary from the expected fit?

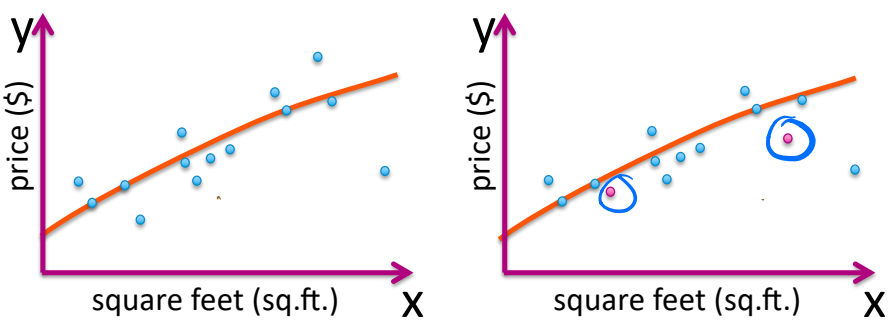


low complexity
→
low variance

©2017 Emily Fox

Variance of high-complexity models

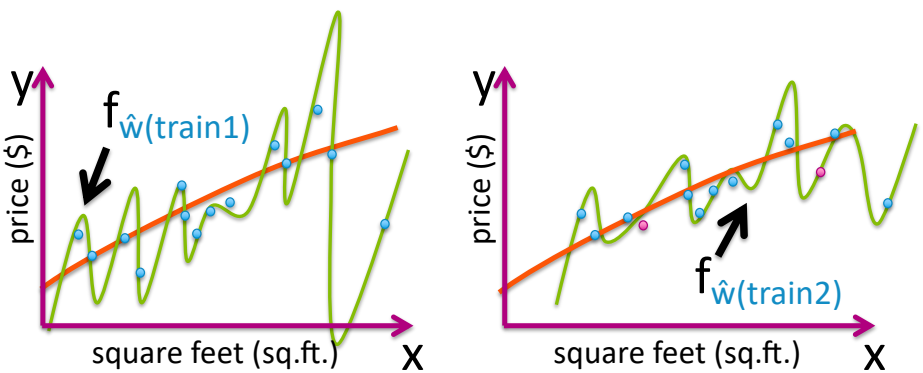
Assume we fit a high-order polynomial



©2017 Emily Fox

Variance of high-complexity models

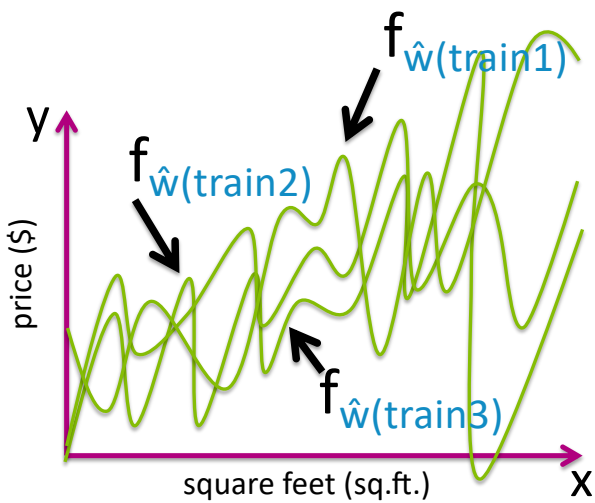
Assume we fit a high-order polynomial



©2017 Emily Fox

Variance of high-complexity models

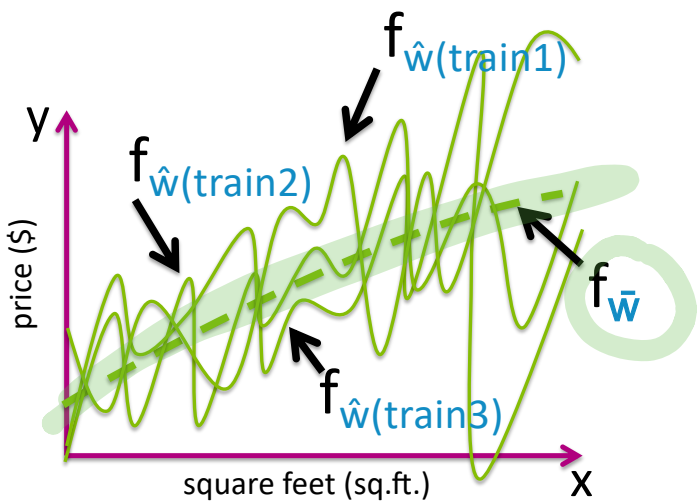
Assume we fit a high-order polynomial



©2017 Emily Fox

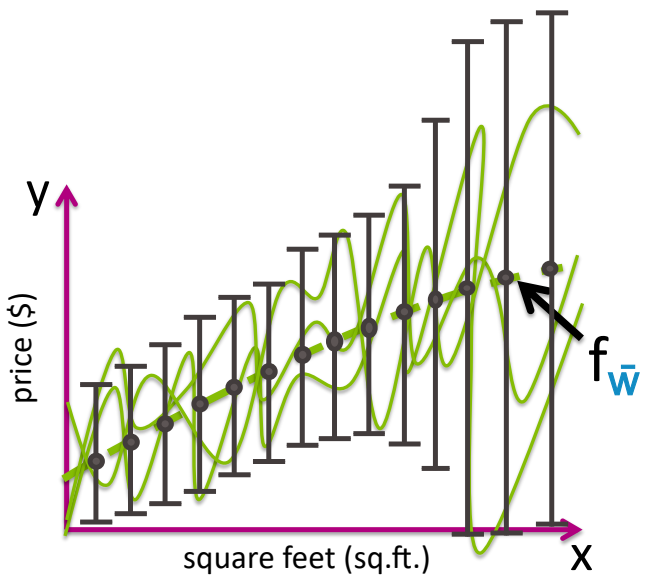
Variance of high-complexity models

Assume we fit a high-order polynomial



©2017 Emily Fox

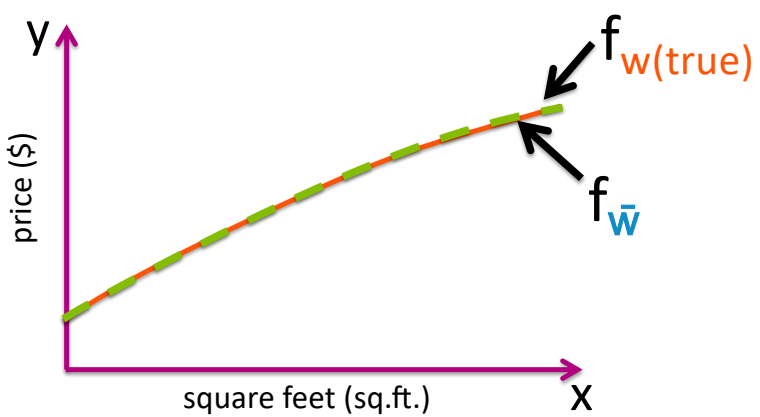
Variance of high-complexity models



high complexity
→
high variance

©2017 Emily Fox

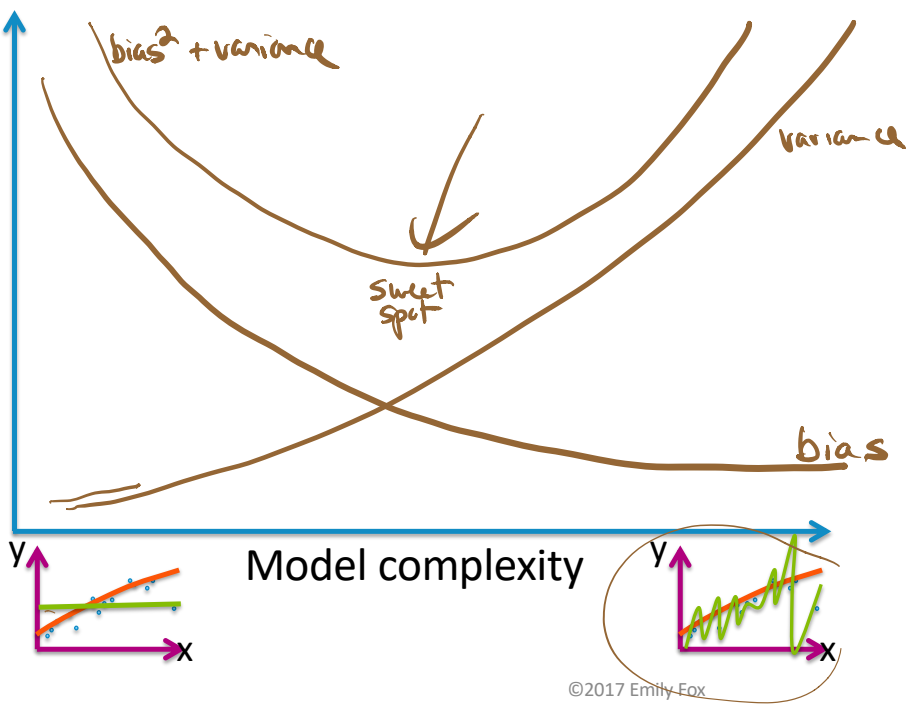
Bias of high-complexity models



high complexity
→
low bias

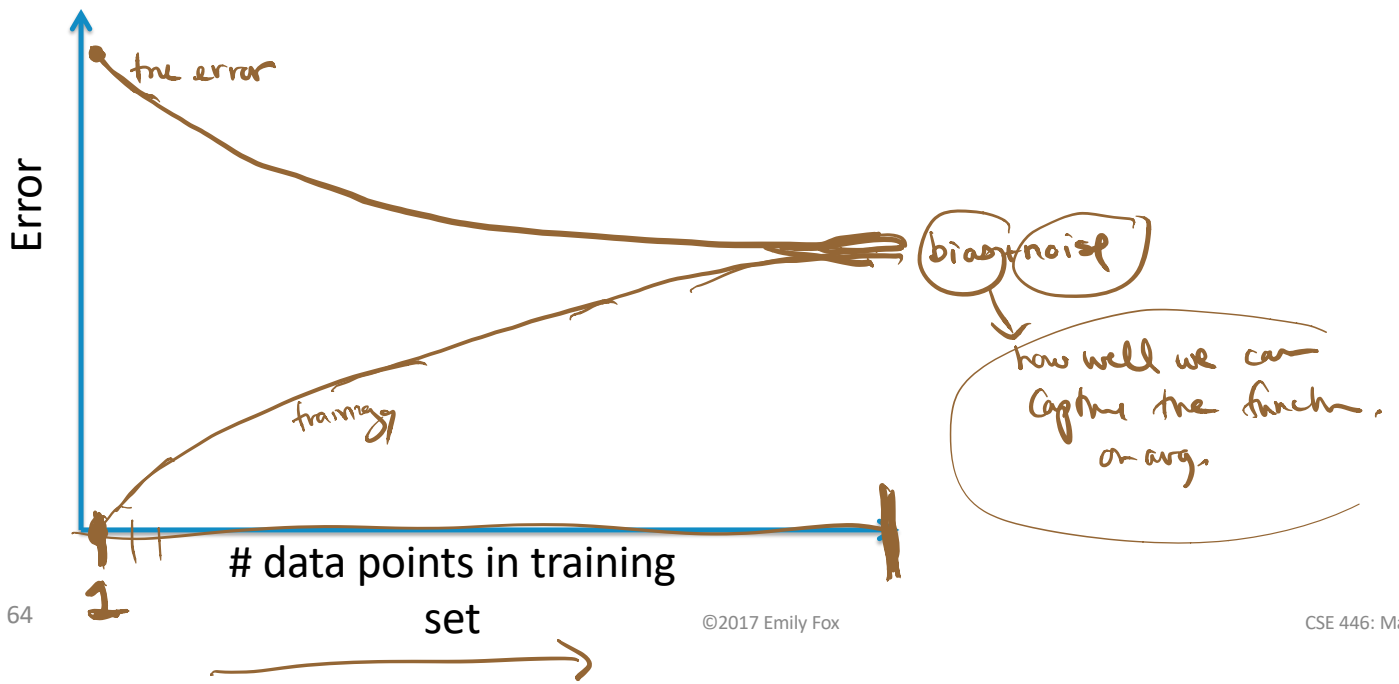
©2017 Emily Fox

Bias-variance tradeoff



©2017 Emily Fox

True error and training error vs. amount of data **for fixed model complexity**



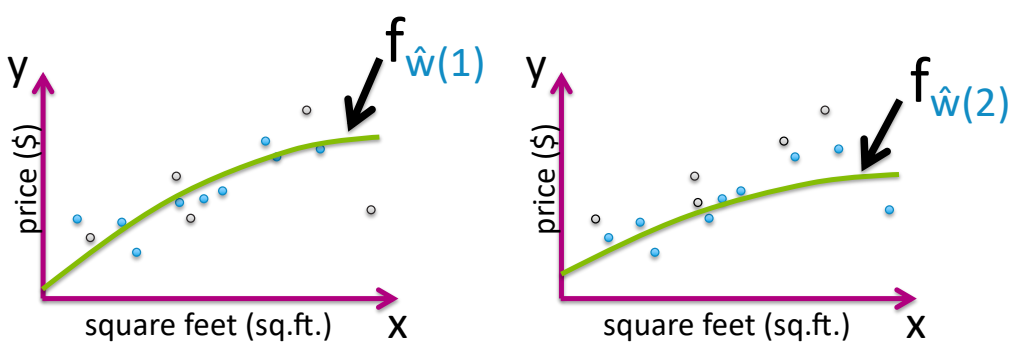
©2017 Emily Fox

Formalize 3 sources of errors…

Accounting for training set randomness

Training set was just a random sample of n houses sold

What if n other houses had been sold and recorded?



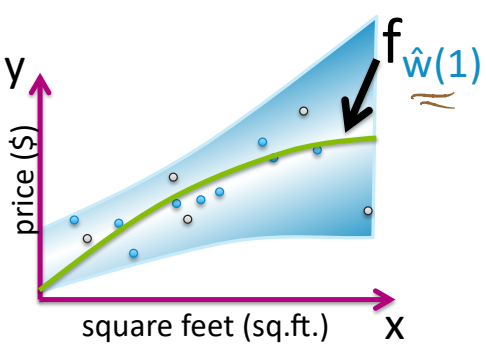
©2017 Emily Fox

Accounting for training set randomness

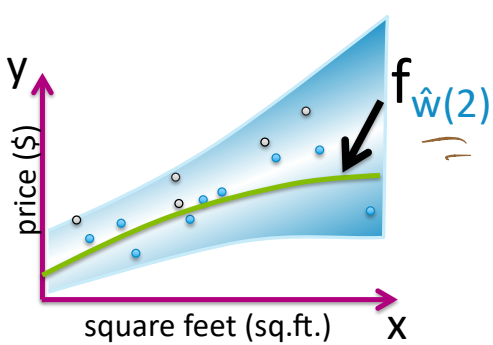
Training set was just a random sample of n houses sold

What if n other houses had been sold and recorded?

generalization error of $\hat{w}(1)$



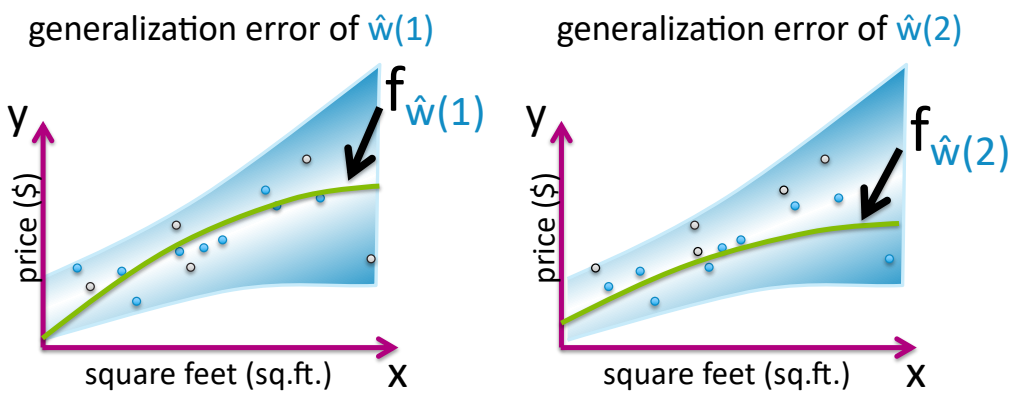
generalization error of $\hat{w}(2)$



©2017 Emily Fox

Accounting for training set randomness

Ideally, want performance averaged over all possible training sets of size n



©2017 Emily Fox

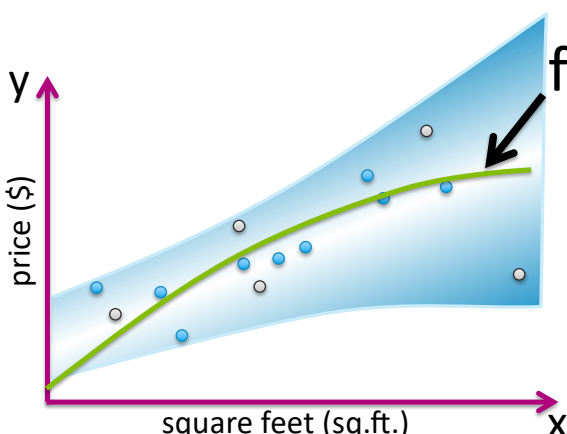
E_{train} $E_{x,y}[(y - f_{\hat{w}(train}(x))^2]$

Expected prediction error

$E_{\text{training set}}$[generalization error of $\hat{w}$(training set)]

averaging over all training sets (weighted by how likely each is)

parameters fit on a specific training set



©2017 Emily Fox

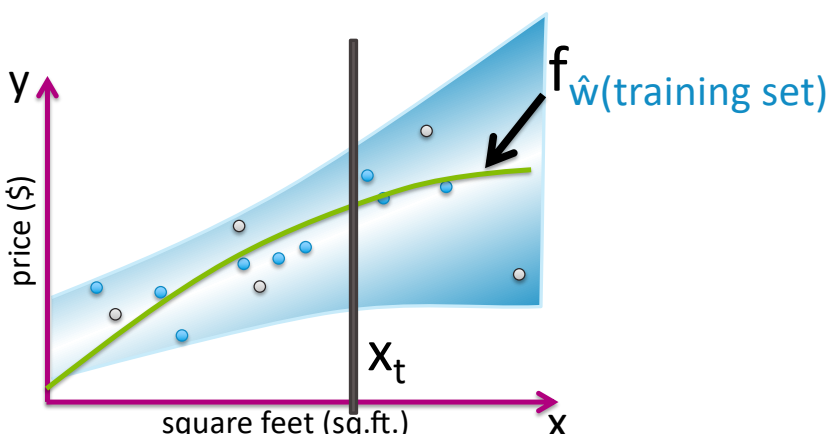
E_{train} $E_{x,y}\left[\left(y - f_{\hat{w}(train)}(x)\right)^2\right]$

$= E_x \; E_{train,\,y|x}\left[\left(y - f_{\hat{w}(train)}(x)\right)^2\right]$

Prediction error at target input

Start by considering:

1. Loss at target x_t (e.g. 2640 sq.ft.)
2. Squared error loss $L(y, f_{\hat{w}}(x_t)) = (y - f_{\hat{w}}(x_t))^2$



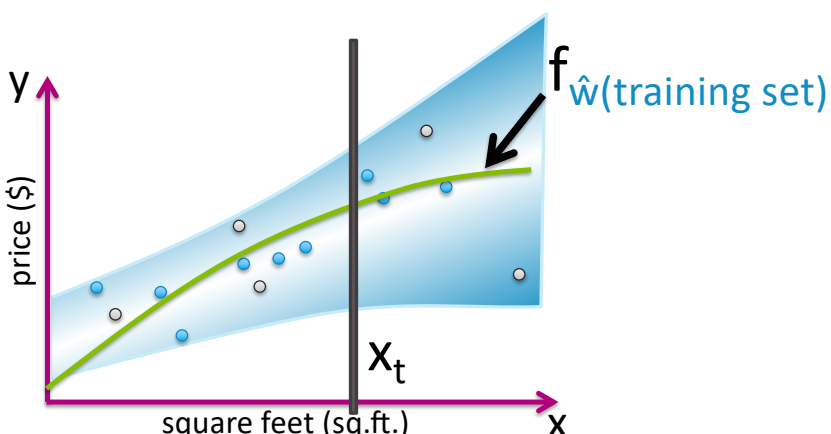
©2017 Emily Fox

Sum of 3 sources of error

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$

Variance of noise
Bias how well on avg model can fit $f_{w(true)}$
Variance variability of fit



©2017 Emily Fox

Error variance of the model

Variance of noise at x_t =

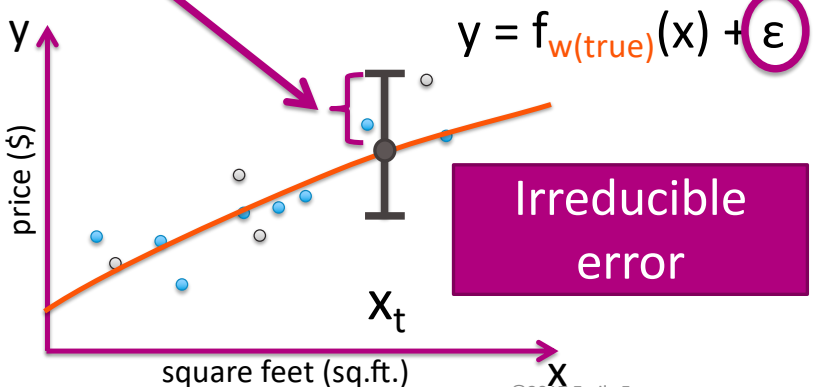
$E_{y|x_t}[(y - f_{w(true)}(x_t))^2] = \sigma^2$

Average prediction error at x_t

$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$

σ^2 = "variance"

$y = f_{w(true)}(x) + \varepsilon$



$f_{w(true)}(x) = E(y|x)$

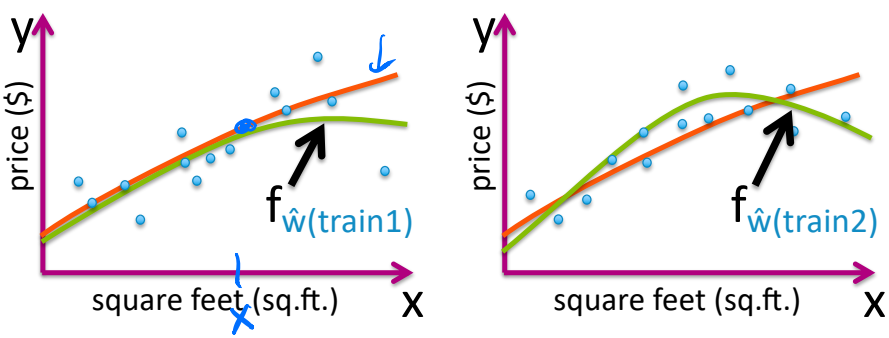
©2017 Emily Fox

Bias of function estimator

Bias how well on avg model can fit $f_{w(true)}$
Variance variability of fit

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$



©2017 Emily Fox

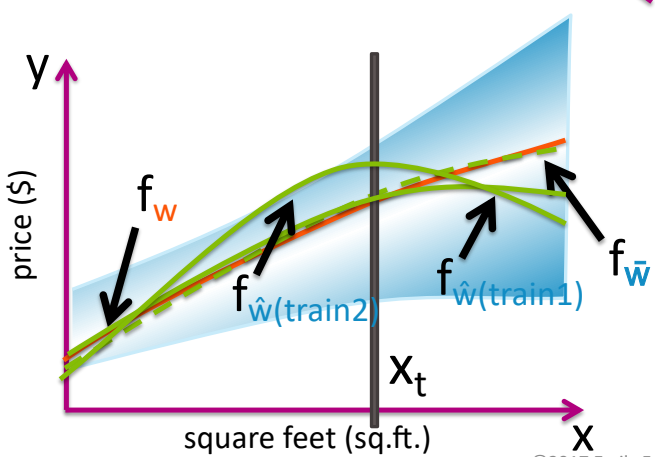
Bias of function estimator

Average estimated function = $f_{\bar{w}}(x)$

True function = $f_w(x)$

$E_{train}[f_{\hat{w}(train)}(x)]$

over all training sets of size N



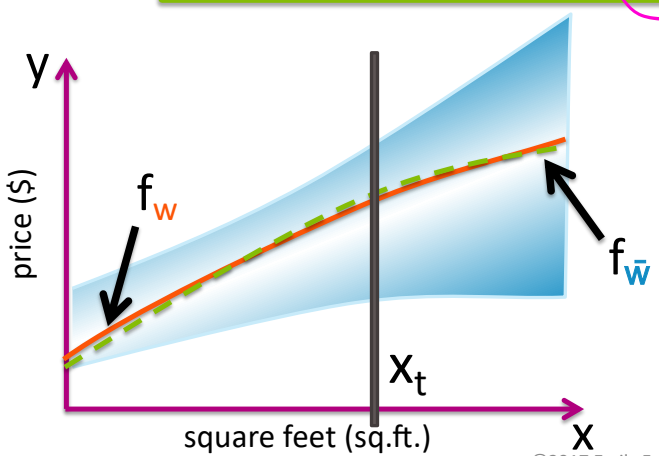
©2017 Emily Fox

Bias of function estimator

Average estimated function = $f_{\bar{w}}(x)$

True function = $f_w(x)$

$$\text{bias}(f_{\hat{w}}(x_t)) = f_w(x_t) - f_{\bar{w}}(x_t)$$



©2017 Emily Fox

Bias of function estimator

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$

Average estimated function = $f_{\bar{w}}(x)$

True function = $f_w(x)$

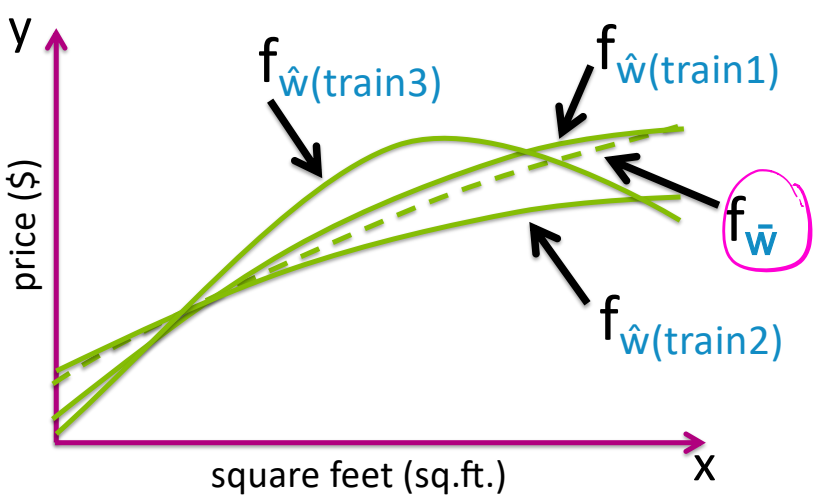
$$\text{bias}(f_{\hat{w}}(x_t)) = f_w(x_t) - f_{\bar{w}}(x_t)$$

©2017 Emily Fox

Variance of function estimator

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$



Bias how well on avg model can fit $f_{w(true)}$

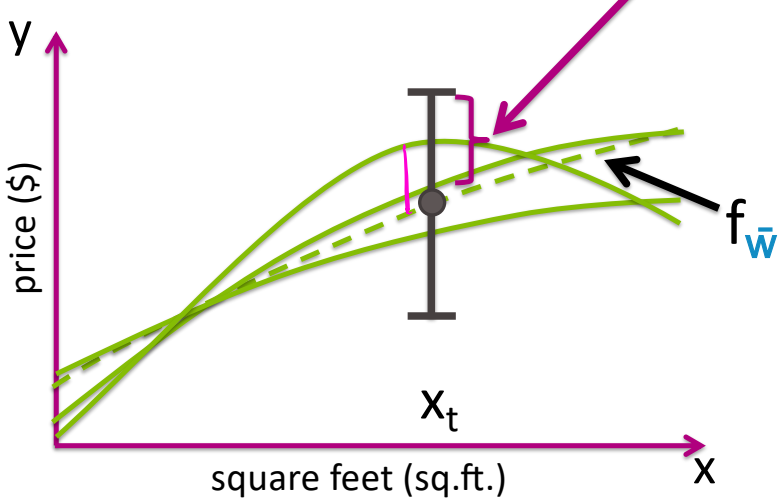
Variance variability of fit

©2017 Emily Fox

Variance of function estimator

Average prediction error at x_t

$$= \sigma^2 + [\text{bias}(f_{\hat{w}}(x_t))]^2 + \text{var}(f_{\hat{w}}(x_t))$$



©2017 Emily Fox

Variance of function estimator

fit on a specific training dataset

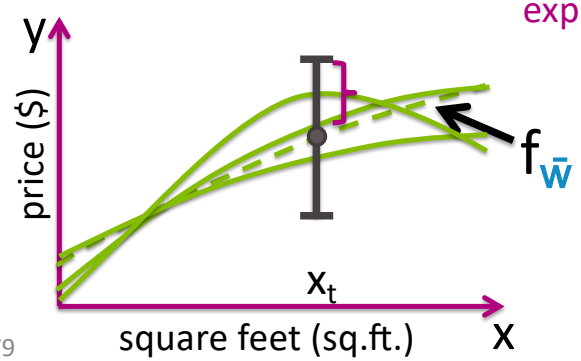
what I expect to learn over all training sets

$$\text{var}(f_{\hat{w}}(x_t)) = E_{\text{train}}[(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t))^2]$$

over all training sets of size N

deviation of specific fit from expected fit at x_t

$f_{\bar{w}}(x_t) = E_{\text{train}}(f_{\hat{w}(\text{train})}(x_t))$



©2017 Emily Fox

Sum of 3 sources of error

Average prediction error at $x_t = E_{\text{train}, y|x}\left[(y - f_{\hat{w}(\text{train})}(x_t))^2\right]$

$= \sigma^2 + \left[\text{bias}(f_{\hat{w}}(x_t))\right]^2 + \text{var}(f_{\hat{w}}(x_t))$

σ^2 at $x_t = E_{y|x_t}\left[(y - f_{w(\text{true})}(x_t))^2\right]$

$\text{Bias}(x_t) = f_{w(\text{true})}(x_t) - f_{\bar{w}}(x_t)$

$\text{Variance}(x_t) = E_{\text{train}}\left[(f_{\hat{w}(\text{train})}(x_t) - f_{\bar{w}}(x_t))^2\right]$

$f_{\bar{w}}(x_t) = E_{\text{train}}\, f[_{\hat{w}(\text{train})}(x_t)]$

Why 3 sources of error?
A formal derivation

Deriving expected prediction error

Expected prediction error

$= E_{train}$ [generalization error of $\hat{w}(train)$]

$= E_{train}\left[E_{x,y}\left[L(y, f_{\hat{w}(train)}(x))\right]\right]$

1. Look at specific x_t
2. Consider $L(y, f_{\hat{w}}(x)) = (y - f_{\hat{w}}(x))^2$

Expected prediction error at x_t

$= E_{train,\ y|x_t}\left[(y - f_{\hat{w}(train)}(x_t))^2\right]$

©2017 Emily Fox

Deriving expected prediction error

Expected prediction error at x_t

$$= E_{train,y|x_t}\left[(y - f_{\hat{w}(train)}(x_t))^2\right]$$

$$= E_{train,y|x_t}\left[((y - f_{w(true)}(x_t)) + (f_{w(true)}(x_t) - f_{\hat{w}(train)}(x_t)))^2\right] = E(c^2) + 2E(cD) + E(D^2)$$

c $\quad$ D

$E(c^2) = E_{y|x_t}\left[(y - f_{w(true)}(x_t))^2\right] = \sigma^2$ noise

$E(cD) = E_{train}\left[(f_{w(true)}(x_t) - f_{\hat{w}(train)}(x_t)) E_{y|x_t}(y - f_{w(true)}(x_t))\right]$

$= 0$ since $f_{w(true)}(x_t) = E(y|x_t)$

$E(X - E(X)) = 0$

$E(D^2) = E_{train}\left[\left(f_{w(true)}(x_t) - f_{\hat{w}(train)}(x_t)\right)^2\right]$ $\quad$ $MSE\left[f_{\hat{w}(train)}(x_t)\right]$

continued on next page.

©2017 Emily Fox

Equating MSE with bias and variance

$$f_{\bar{w}}(x) = E_{train}\, f[\hat{w}_{(train)}(x)]$$

$$E((A+B)^2) = E(A^2) + 2E(AB) + E(B^2)$$

$$MSE\left[f_{\hat{w}(train)}(x_t)\right]$$

$$= E_{train}\left[\left(f_{w(true)}(x_t) - f_{\hat{w}(train)}(x_t)\right)^2\right]$$

$$= E_{train}\left[\left(\underbrace{\left(f_{w(true)}(x_t) - f_{\bar{w}}(x_t)\right)}_{A} + \underbrace{\left(f_{\bar{w}}(x_t) - f_{\hat{w}(train)}(x_t)\right)}_{B}\right)^2\right]$$

$$E_{train}(A^2) = E_{train}\left[\left(f_{w(true)}(x_t) - f_{\bar{w}}(x_t)\right)^2\right] = \left(f_{w(true)}(x_t) - f_{\bar{w}}(x_t)\right)^2 = \left(bias(x_t)\right)^2$$

$$E_{train}(AB) = \left(f_{w(true)}(x_t) - f_{\bar{w}}(x_t)\right) E_{train}\left[f_{\bar{w}}(x_t) - f_{\hat{w}(train)}(x_t)\right]$$

$= 0$ since $E_{train}\left(f_{\hat{w}(train)}(x_t)\right) = f_{\bar{w}}(x_t)$

$$E_{train}(B^2) = E_{train}\left[\left(f_{\hat{w}(train)}(x_t) - f_{\bar{w}}(x_t)\right)^2\right] = \text{variance of } f_{\hat{w}}(x_t)$$

©2017 Emily Fox

Putting it all together

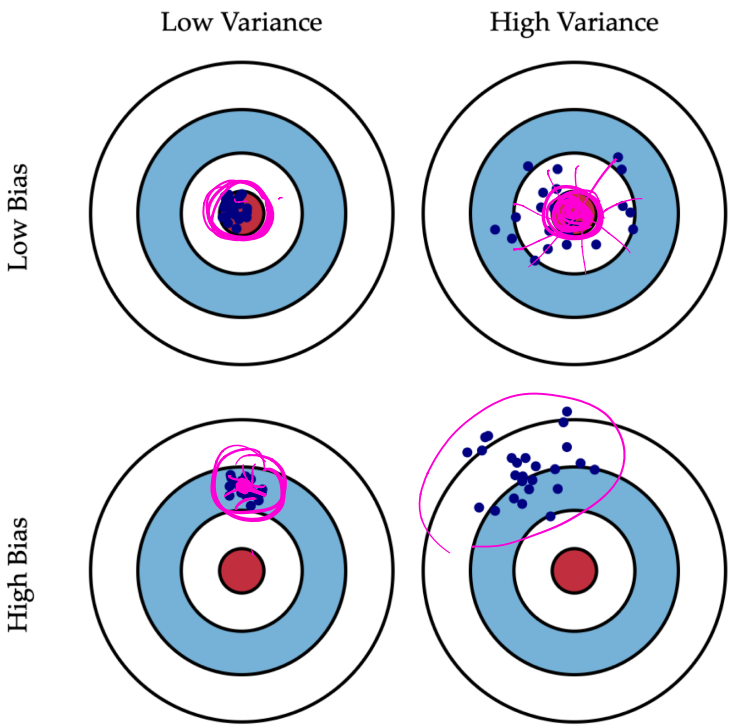
Expected prediction error at x_t

$= \sigma^2 + \text{MSE}\left[f_{\hat{w}}(x_t)\right]$

$= \sigma^2 + \left[\text{bias}(f_{\hat{w}}(x_t))\right]^2 + \text{var}(f_{\hat{w}}(x_t))$

3 sources of error

©2017 Emily Fox



Courtesy of Scott Fortmann-Roe

Summary of assessing performance

What you can do now…

- Describe what a loss function is and give examples
- Contrast training, generalization, and test error
- Compute training and test error given a loss function
- Discuss issue of assessing performance on training set
- Describe tradeoffs in forming training/test splits
- Understand the 3 sources of avg. prediction error
  - Irreducible error, bias, and variance

©2017 Emily Fox