Regression:

Predicting House Prices

STAT/CSE 416: Intro to Machine Learning
Emily Fox
University of Washington
March 29, 2018

How much is my house worth?

Data

input *output*

(x_1 = sq.ft., y_1 = \$)

(x_2 = sq.ft., y_2 = \$)

(x_3 = sq.ft., y_3 = \$)

(x_4 = sq.ft., y_4 = \$)

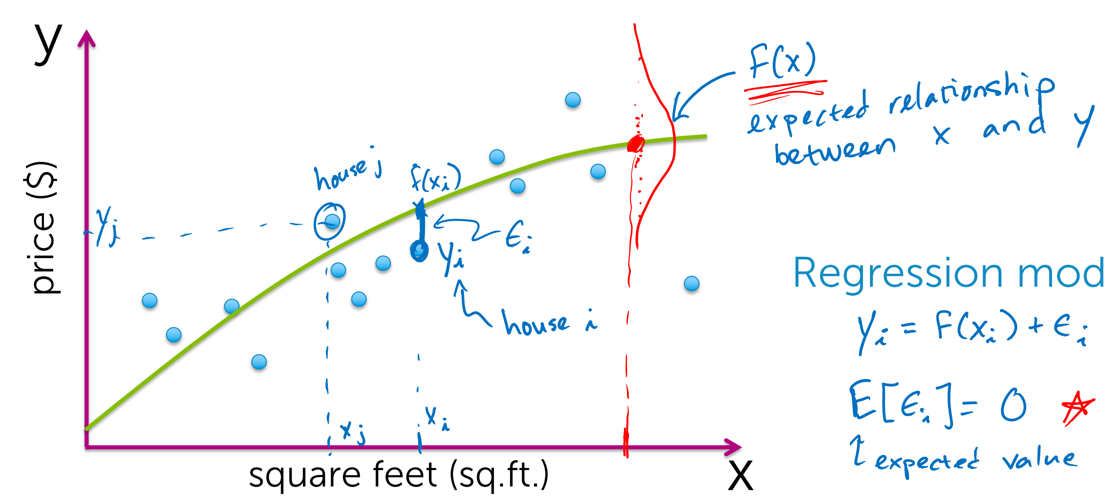
(x_5 = sq.ft., y_5 = \$)

⋮

Input vs. Output:

- y is the quantity of interest
- assume y can be predicted from x

Model – How we *assume* the world works

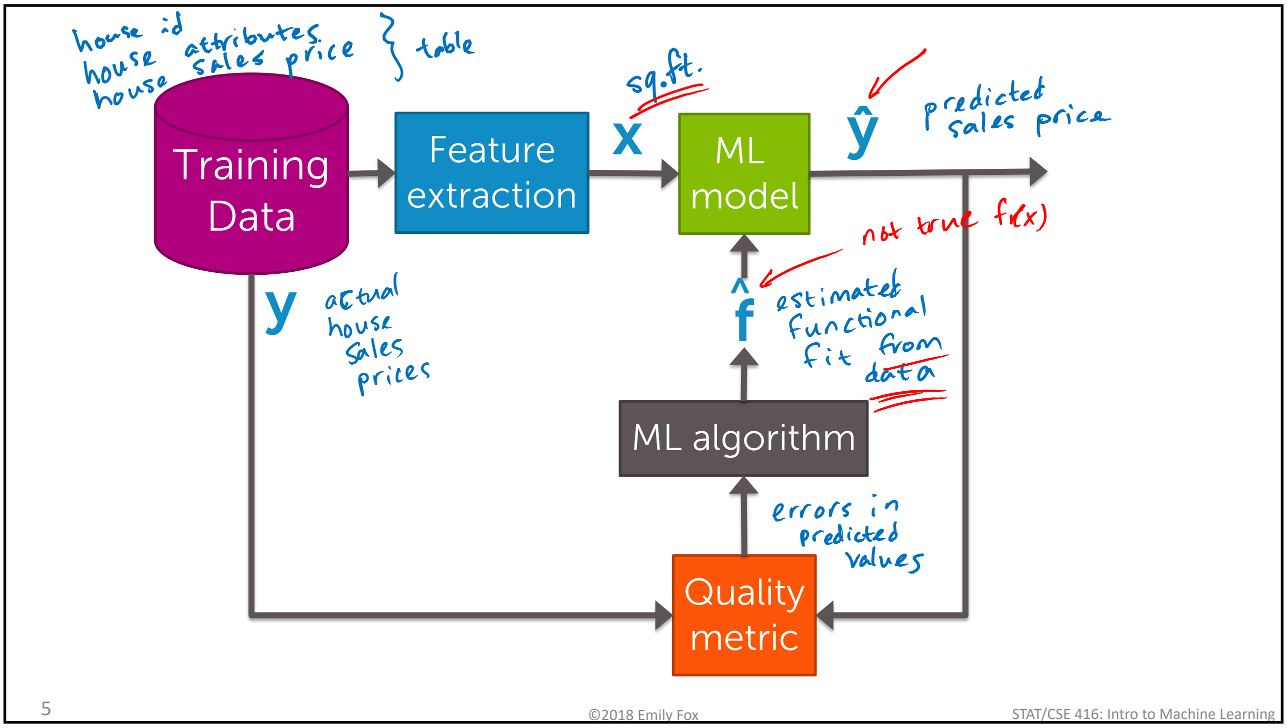


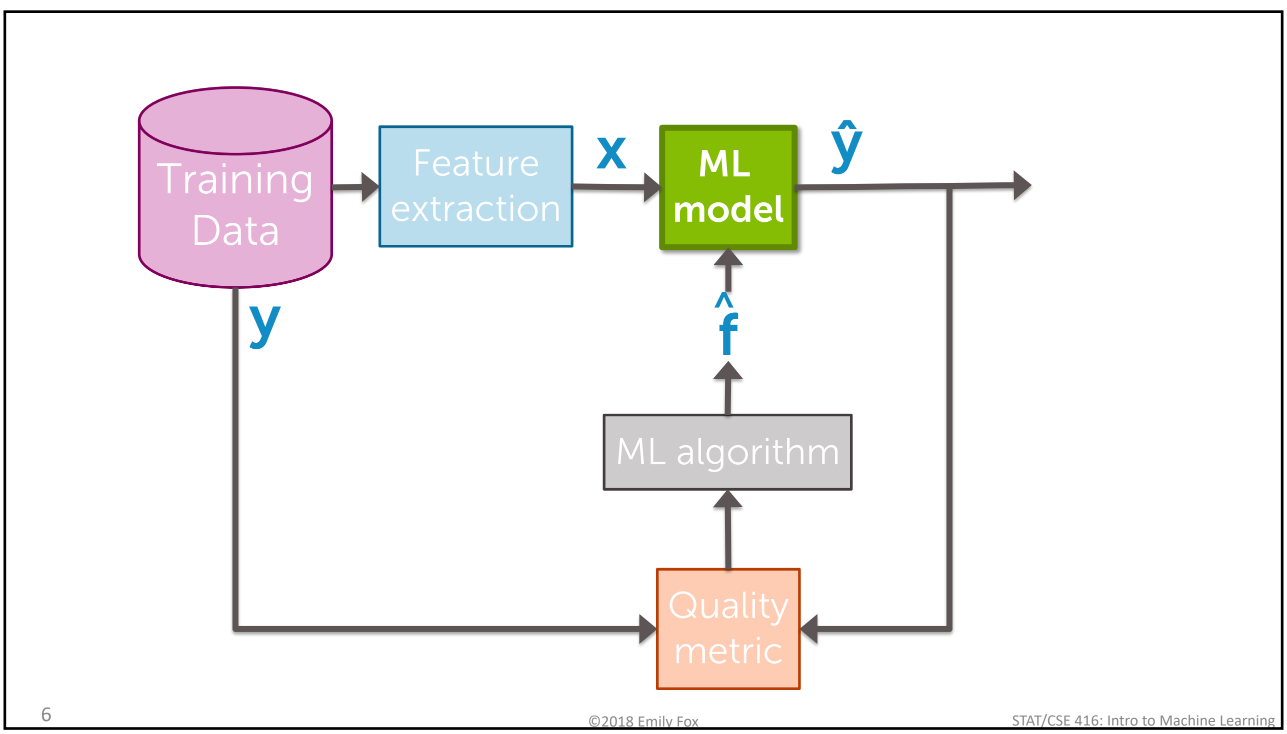
Regression model:

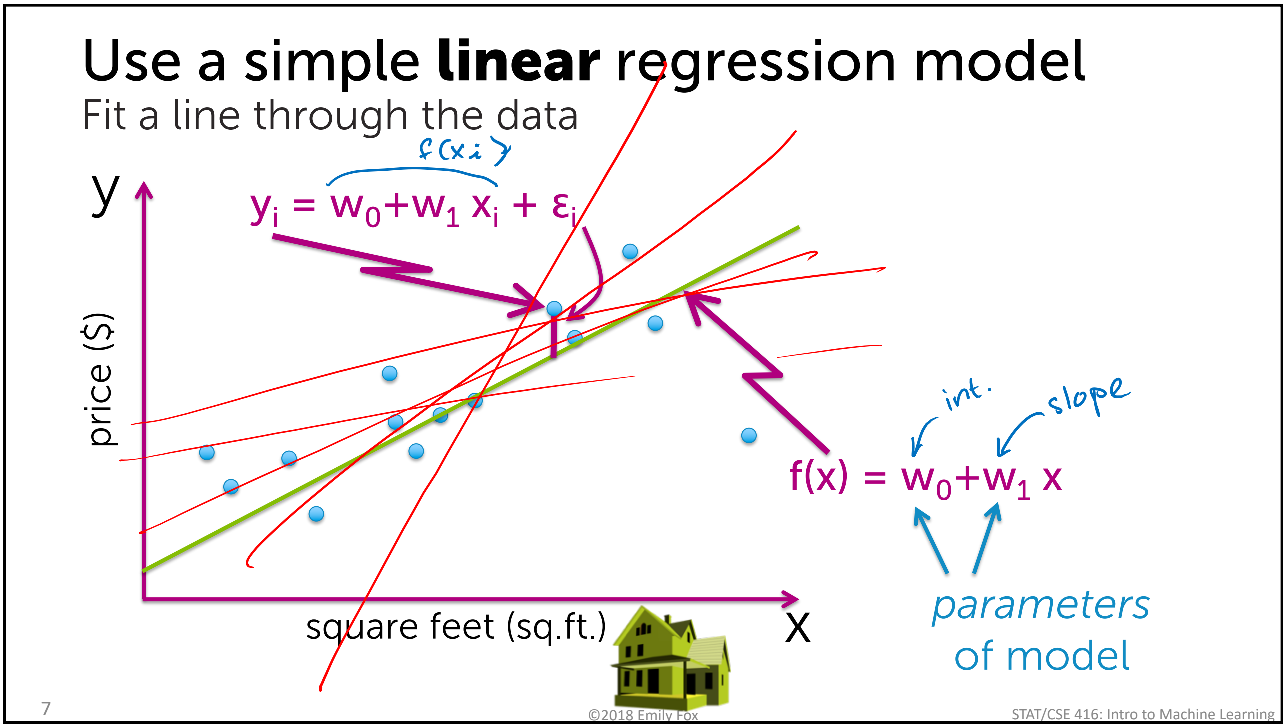
$y_i = F(x_i) + \epsilon_i$

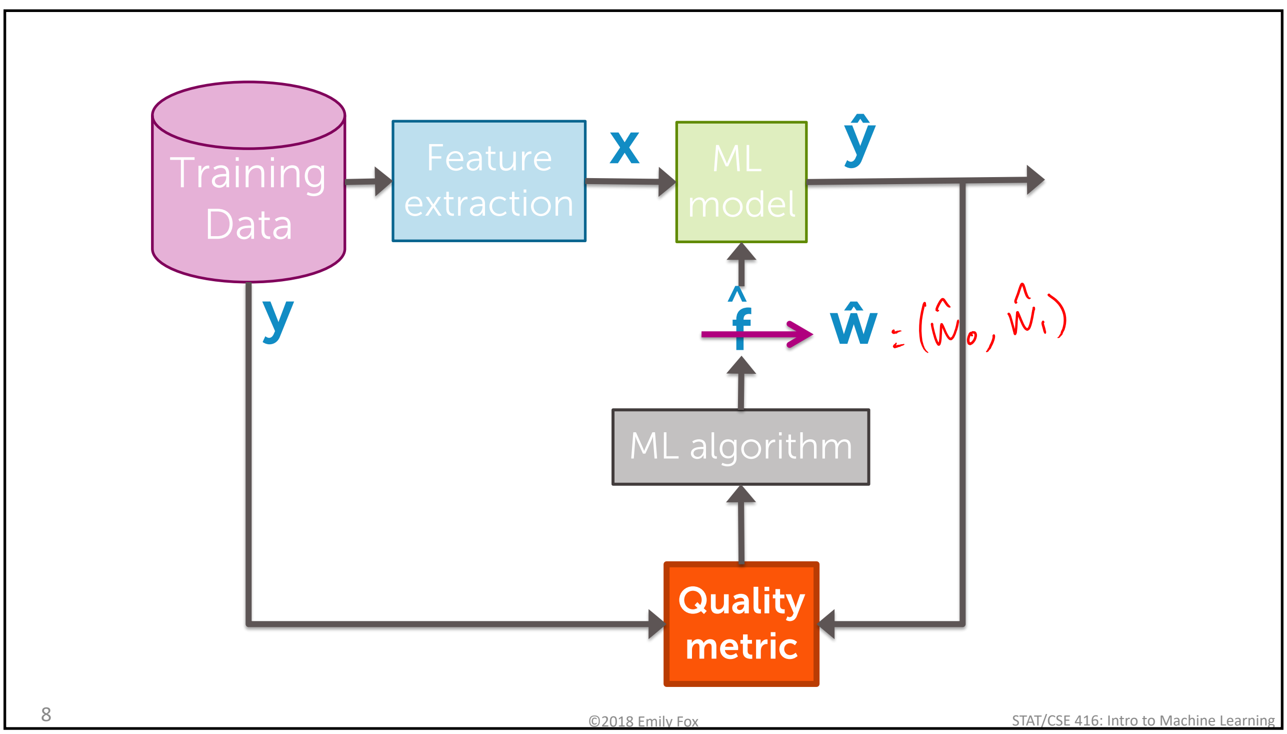
$E[\epsilon_i] = 0$ ☆

↑ expected value


house id
house attributes
house sales price
table
Training Data
Feature extraction
x
sq.ft.
ML model
ŷ
predicted sales price
not true f(x)
y
actual house sales prices
f̂
estimated functional fit from data
ML algorithm
errors in predicted values
Quality metric
5
©2018 Emily Fox
STAT/CSE 416: Intro to Machine Learning


Training Data
Feature extraction
x
ML model
ŷ
y
f̂
ML algorithm
Quality metric
6
©2018 Emily Fox
STAT/CSE 416: Intro to Machine Learning


Use a simple linear regression model
Fit a line through the data
f(xi)
yi = w0+w1 xi + εi
y
price ($)
square feet (sq.ft.)
x
int.
slope
f(x) = w0+w1 x
parameters of model
7
©2018 Emily Fox
STAT/CSE 416: Intro to Machine Learning


Training Data
Feature extraction
x
ML model
ŷ
y
f̂
ŵ = (ŵ0, ŵ1)
ML algorithm
Quality metric
8
©2018 Emily Fox
STAT/CSE 416: Intro to Machine Learning

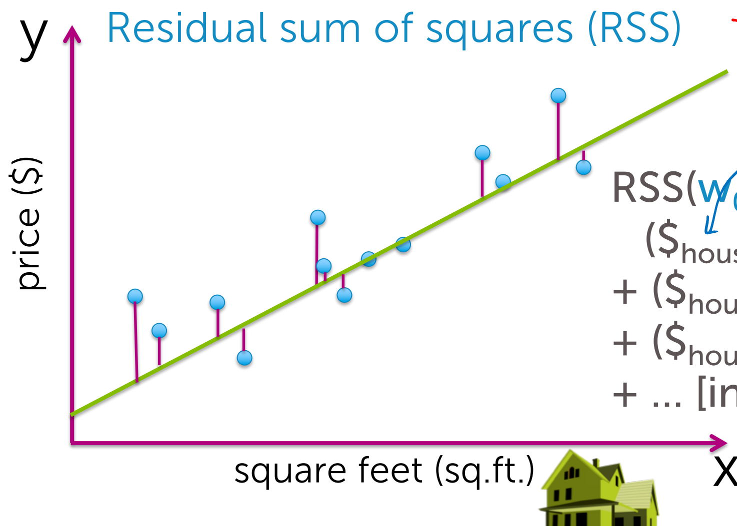
"Cost" of using a given line

$RSS(w_0,w_1) = \sum_{i=1}^{N} (y_i - [w_0+w_1x_i])^2$

$\sum_{i=1}^{N} a_i = a_1 + a_2 + \dots + a_N$

$a_i = (y_i - [w_0+w_1x_i])^2$

Residual sum of squares (RSS)



actual

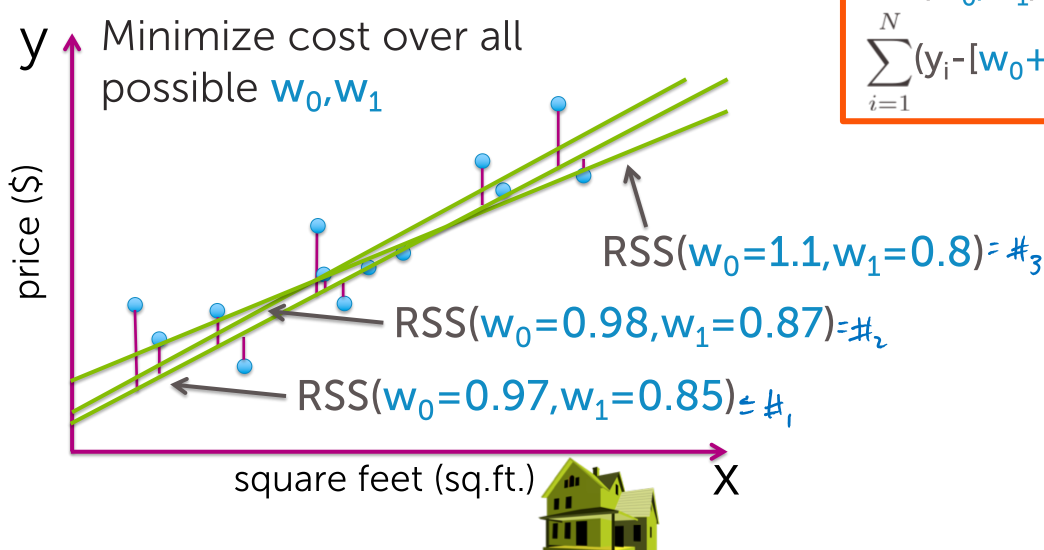
point on the line

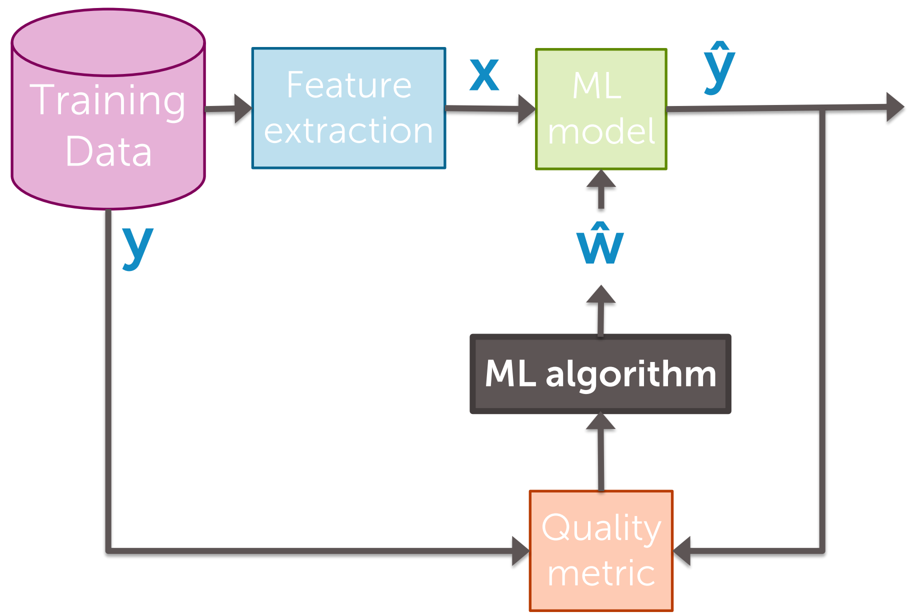
RSS(w_0,w_1) =
 (\$house 1-[w_0+$w_1$sq.ft.house 1])²
+ (\$house 2-[w_0+$w_1$sq.ft.house 2])²
+ (\$house 3-[w_0+$w_1$sq.ft.house 3])²
+ … [include all houses]

Find "best" line

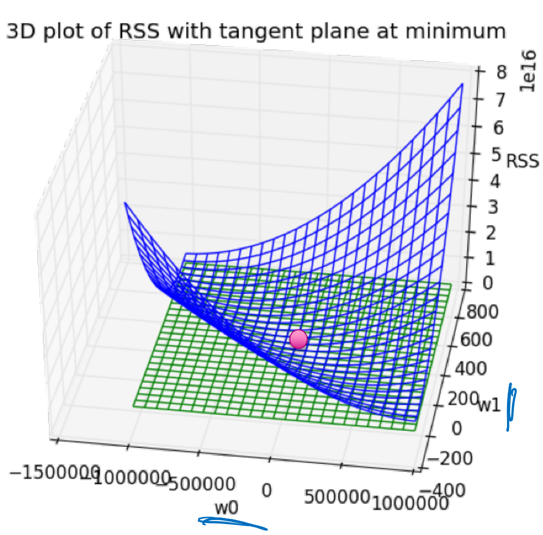
Minimize cost over all possible w_0,w_1

$$RSS(w_0,w_1) = \sum_{i=1}^{N}(y_i-[w_0+w_1x_i])^2$$





Minimizing the cost

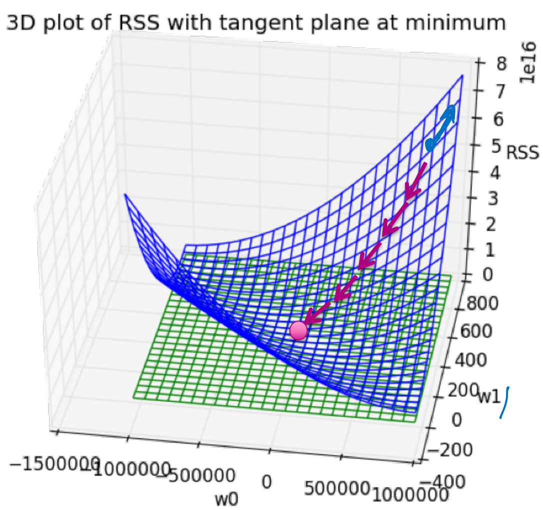


Minimize function over all possible w_0, w_1

$$\min_{w_0, w_1} \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

$RSS(w_0, w_1)$ is a function of 2 variables

Gradient descent



Algorithm:

while not converged

$w^{(t+1)} \leftarrow w^{(t)} - \eta \nabla RSS(w^{(t)})$

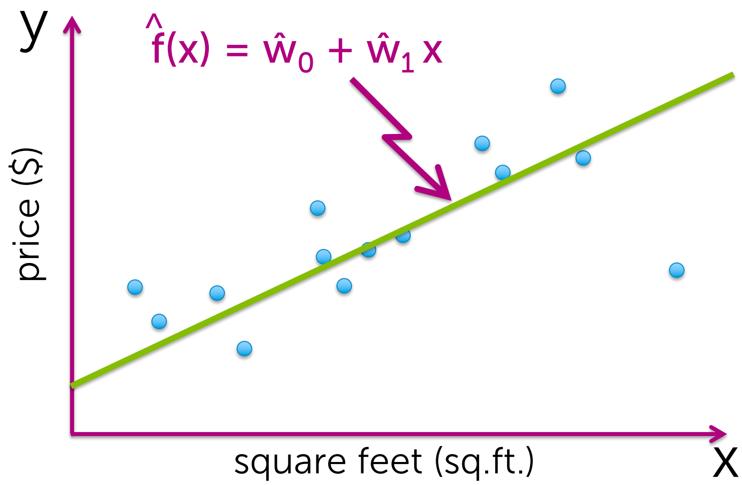
$\hat{w}$

stepsize

best guess for this dataset

The fitted line: use + interpretation

Model vs. fitted line



Regression model:

$y_i = w_0 + w_1 x_i + \varepsilon_i$

model params (unknown variables)

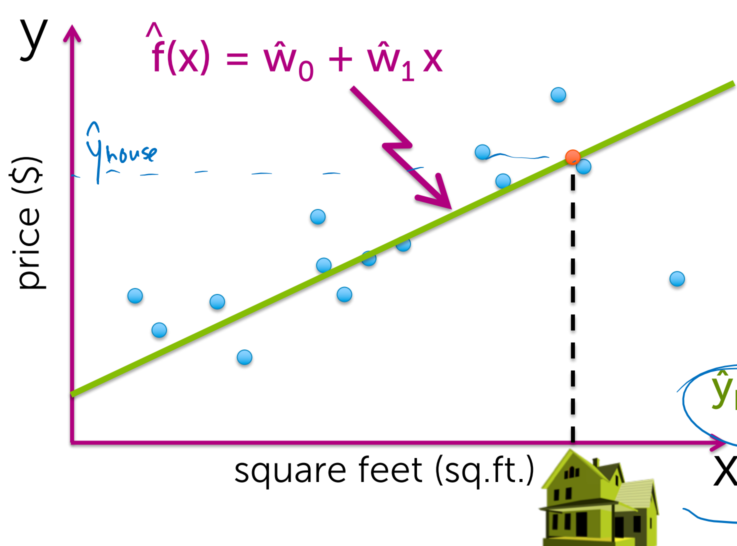
Estimated parameters:

$\hat{w}_0 = -44850$, $\hat{w}_1 = 280.76$

estimated values

15 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Seller:
Predicting your house price



Regression model:

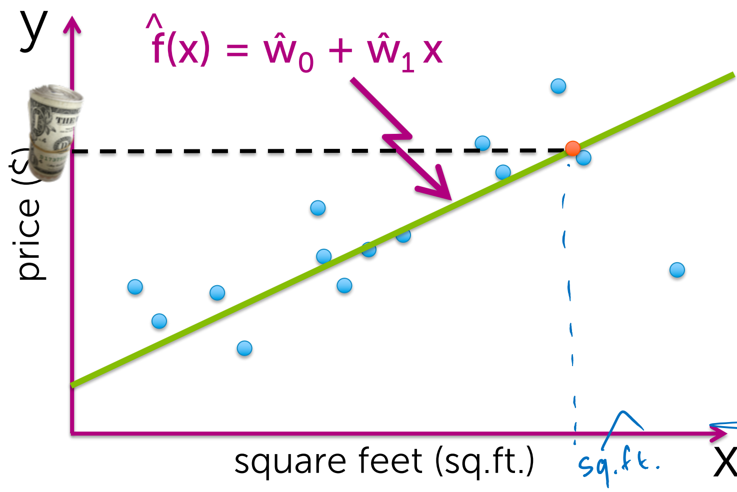
$y_i = w_0 + w_1 x_i + \varepsilon_i$

Best guess of your house price:

$\hat{y}_{house} = \hat{w}_0 + \hat{w}_1 \, sq.ft._{house}$

16 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Buyer:
Predicting size of house



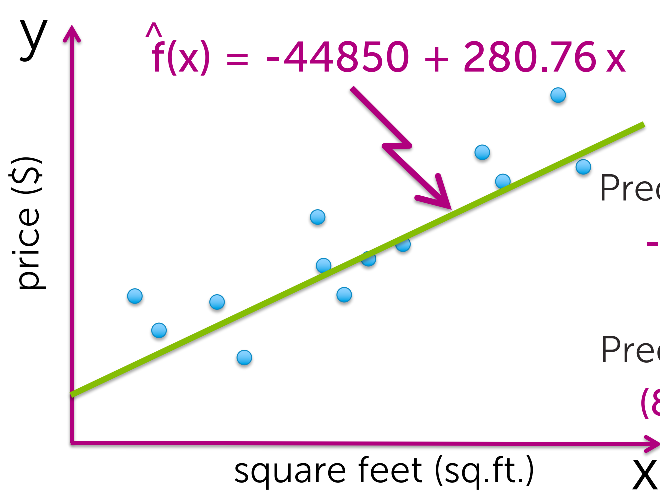
Regression model:

$y_i = w_0 + w_1 x_i + \varepsilon_i$

Best guess of size of house you can afford:

$\$_{\text{in bank}} = \hat{w}_0 + \hat{w}_1 \widehat{\text{sq.ft.}}$

A concrete example



Predict \$ of 2,640 sq.ft. house:

-44850 + 280.76 * 2,640

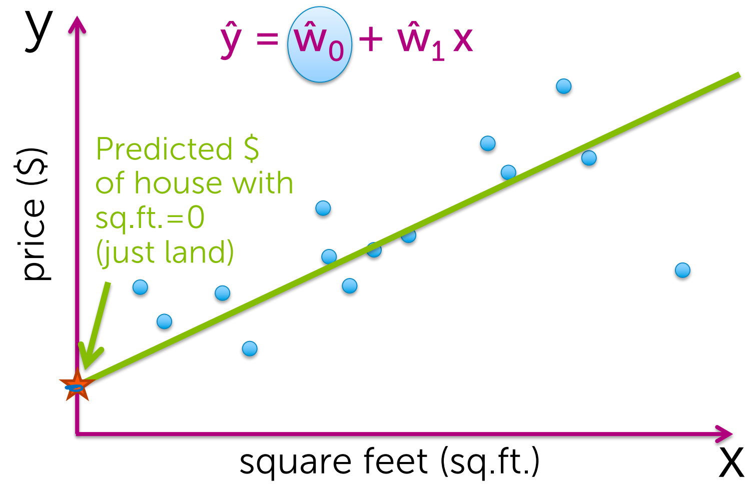
= \$696,356

Predict sq.ft. of \$859,000 sale:

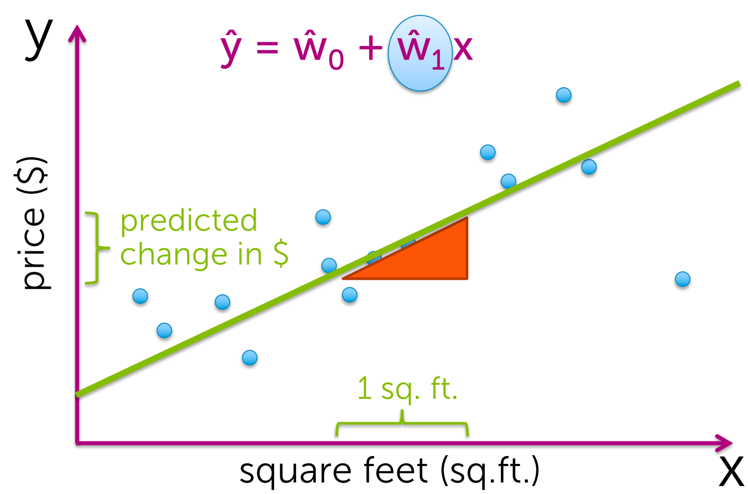
(859000+44850)/ 280.76

= 3,219 sq.ft.

Interpreting the coefficients



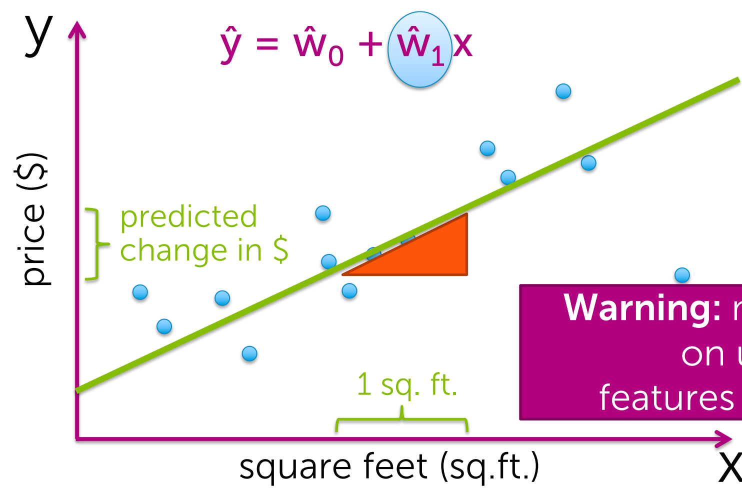
Interpreting the coefficients



predicted change in $ per unit change in sq.ft.

Interpreting the coefficients

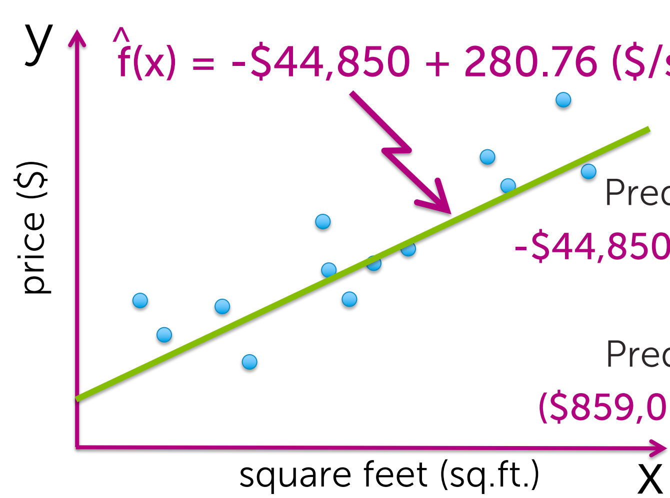
$\hat{y} = \hat{w}_0 + \hat{w}_1 x$



Warning: magnitude depends on units of both features and observations

A concrete example

$\hat{f}(x)$ = -$44,850 + 280.76 ($/sq.ft.) x



Predict $ of 2,640 sq.ft. house:

-$44,850 + 280.76 ($/sq.ft.) * 2,640 sq.ft.

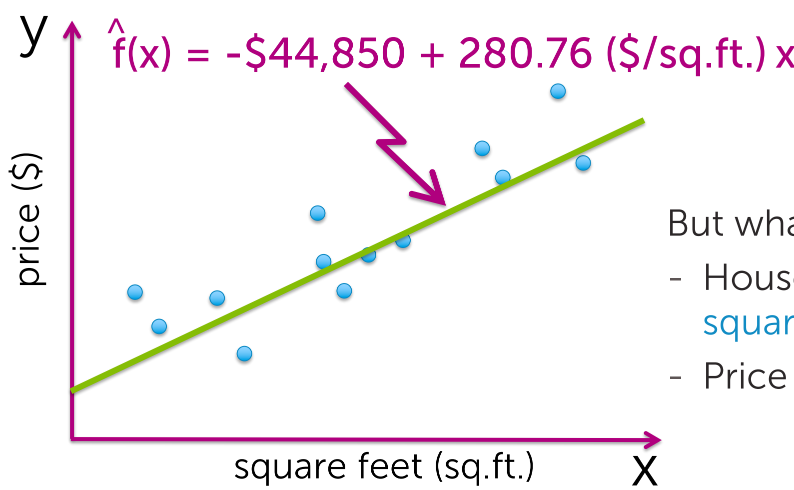
= $696,356

Predict sq.ft. of $859,000 sale:

($859,000+$44,850)/ 280.76 ($/sq.ft.)

= 3,219 sq.ft.

A concrete example



But what if:

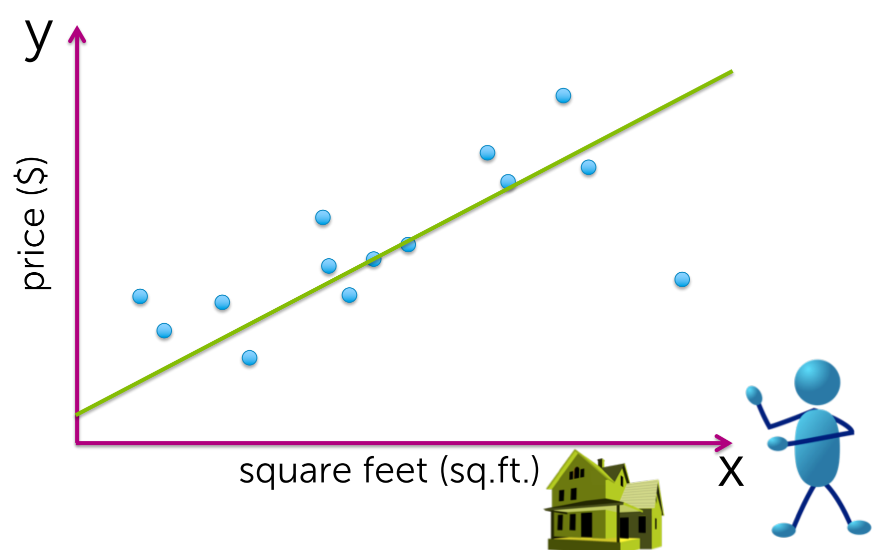
- House was measured in square meters?
- Price was measured in RMB?

23 ©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Adding higher order effects

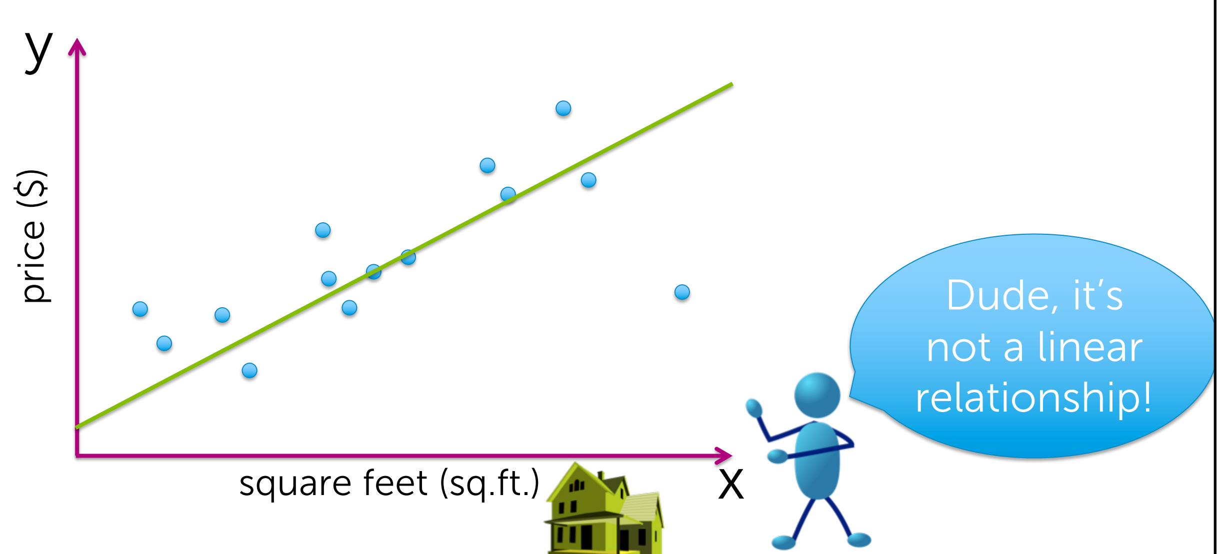
©2018 Emily Fox STAT/CSE 416: Intro to Machine Learning

Fit data with a line or … ?

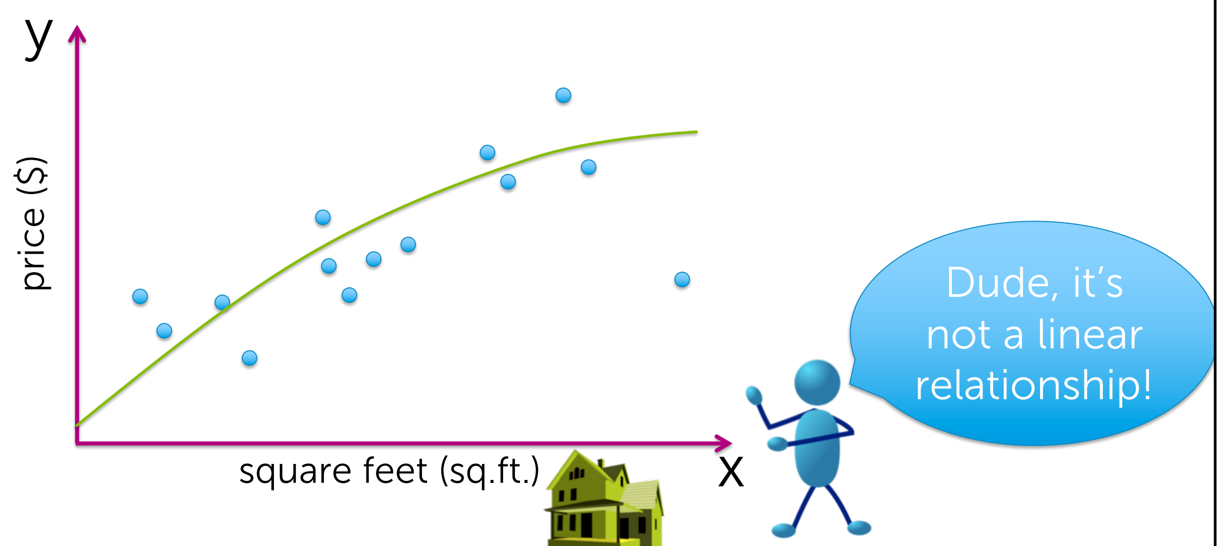


You show your friend your analysis

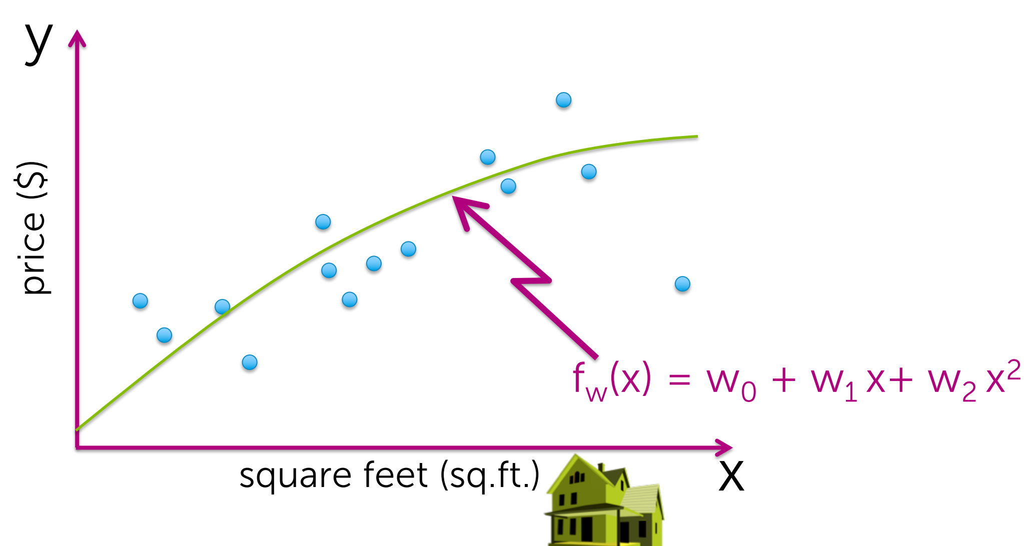
Fit data with a line or … ?



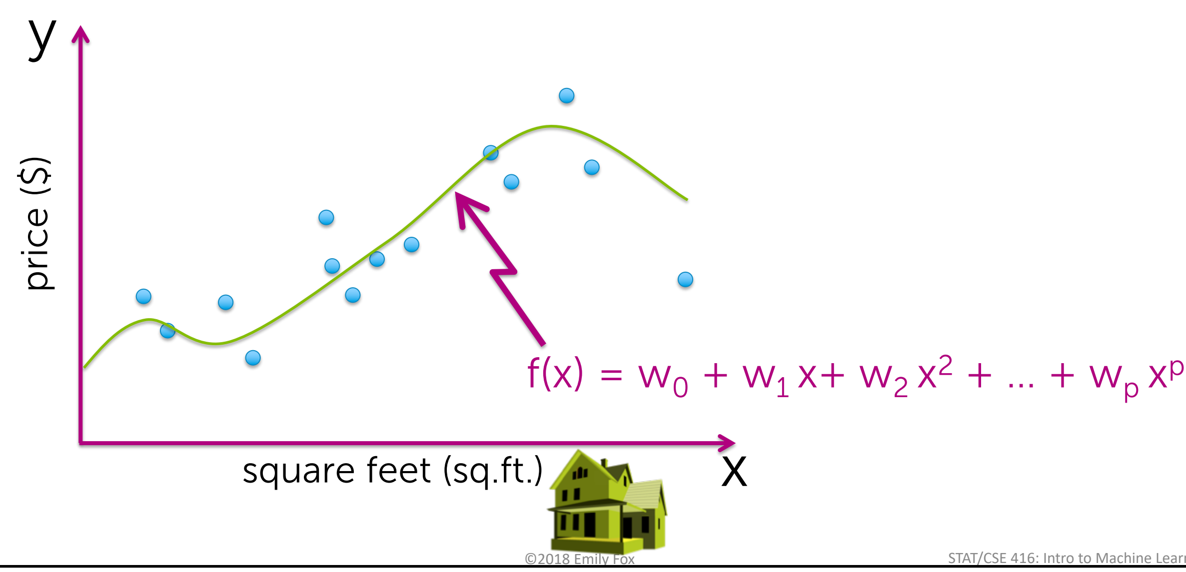
What about a quadratic function?



What about a quadratic function?



Even higher order polynomial



Polynomial regression

Model:

$y_i = w_0 + w_1 x_i + w_2 x_i^2 + \ldots + w_p x_i^p + \varepsilon_i$

treat as different **features**

feature 1 = 1 (constant) *parameter* 1 = w_0

feature 2 = x *parameter* 2 = w_1

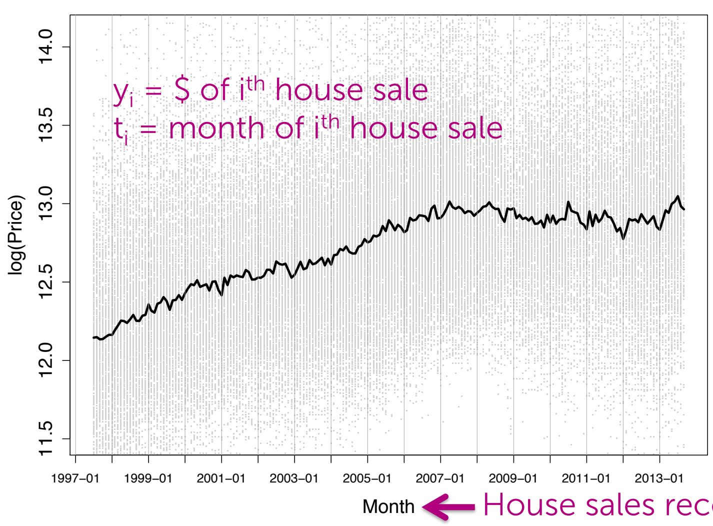
feature 3 = x^2 *parameter* 3 = w_2

… …

feature p+1 = x^p *parameter* p+1 = w_p

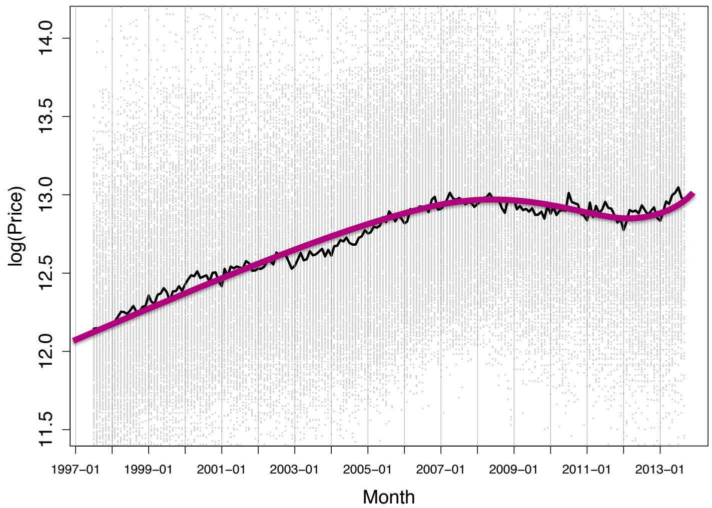
Other functions of one input

Motivating application: Detrending time series



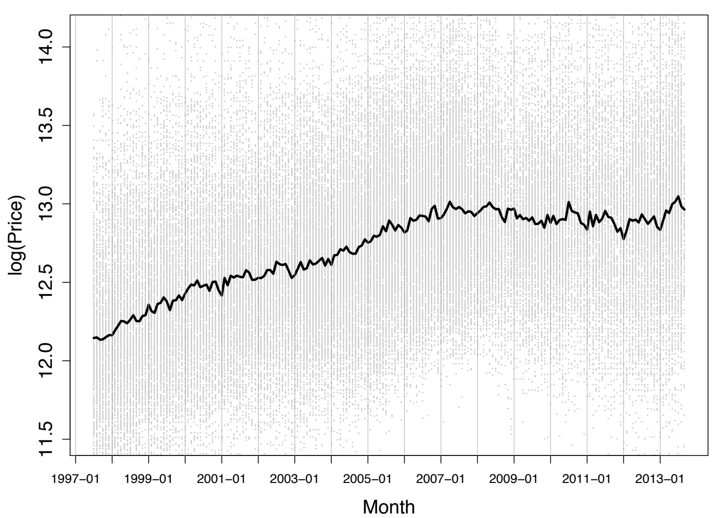
House sales recorded monthly

Trends over time



On average, house prices tend to increase with time

Seasonality



Most houses listed in summer
+
Good houses sell quickly

Few homes listed in Nov./Dec.
+
Transactions often leftover inventory or special circumstances

An example detrending

Model:

$y_i = w_0 + w_1 t_i + w_2 \sin(2\pi t_i / 12 - \Phi) + \varepsilon_i$

Linear trend

Seasonal component = Sinusoid with period 12 (resets annually)

Unknown phase/shift

Trigonometric identity: $\sin(a-b)=\sin(a)\cos(b)-\cos(a)\sin(b)$

→ $\sin(2\pi t_i / 12 - \Phi) = \sin(2\pi t_i / 12)\cos(\Phi) - \cos(2\pi t_i / 12)\sin(\Phi)$

An example detrending

Equivalently,

$y_i = w_0 + w_1 t_i + w_2 \sin(2\pi t_i / 12) + w_3 \cos(2\pi t_i / 12) + \varepsilon_i$

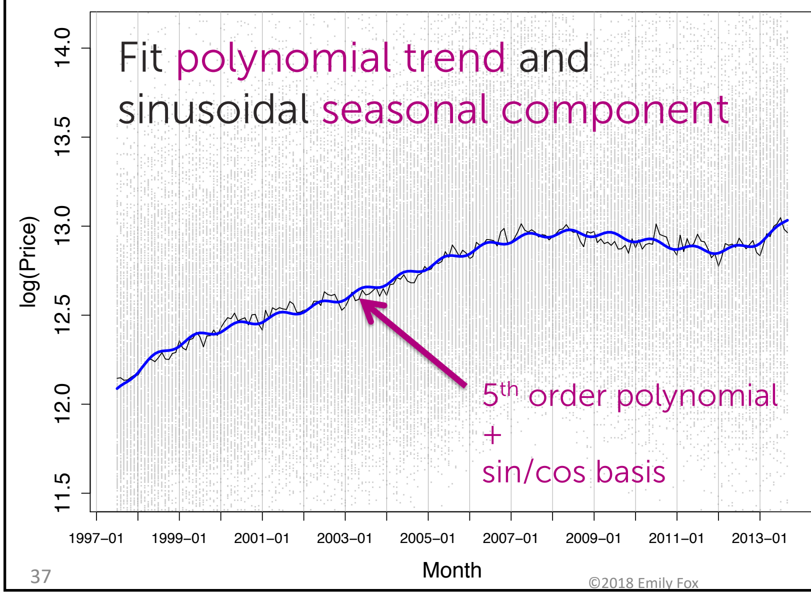
feature 1 = 1 (constant)

feature 2 = t

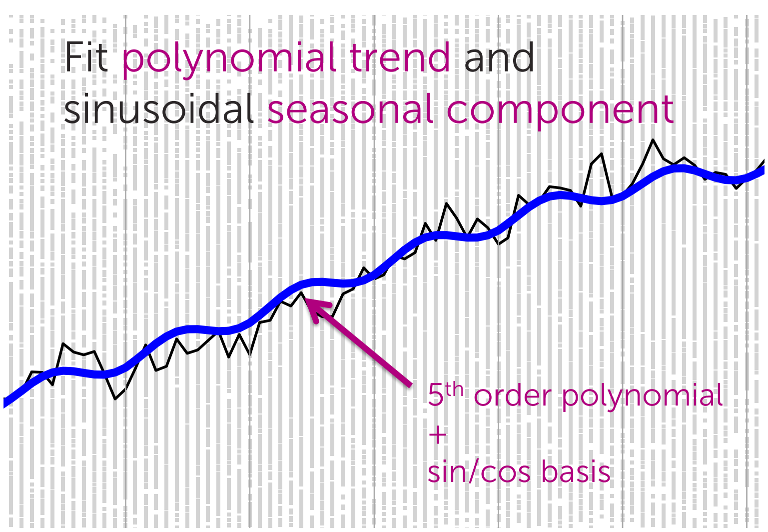
feature 3 = $\sin(2\pi t/12)$

feature 4 = $\cos(2\pi t/12)$

Detrended housing data

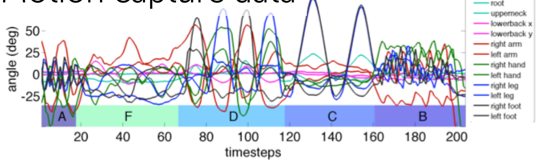


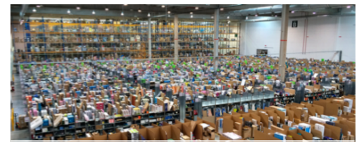
Zoom in…



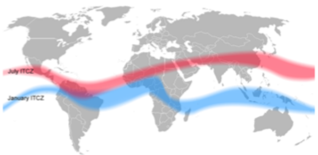
Other examples of seasonality

Motion capture data

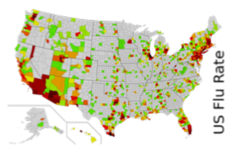




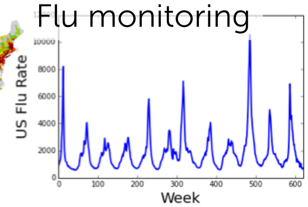
Demand forecasting
(e.g., jacket purchases)



Weather modeling
(e.g., temperature, rainfall)



Flu monitoring



Generic basis expansion

Model:

$$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \varepsilon_i$$

$$= \sum_{j=0}^{D} w_j h_j(x_i) + \varepsilon_i$$

j^{th} *feature*

j^{th} *regression coefficient or weight*

Generic basis expansion

Model:

$y_i = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i) + \varepsilon_i$

$$= \sum_{j=0}^{D} w_j h_j(x_i) + \varepsilon_i$$

feature 1 = $h_0(x)$…often 1 (constant)

feature 2 = $h_1(x)$… e.g., x

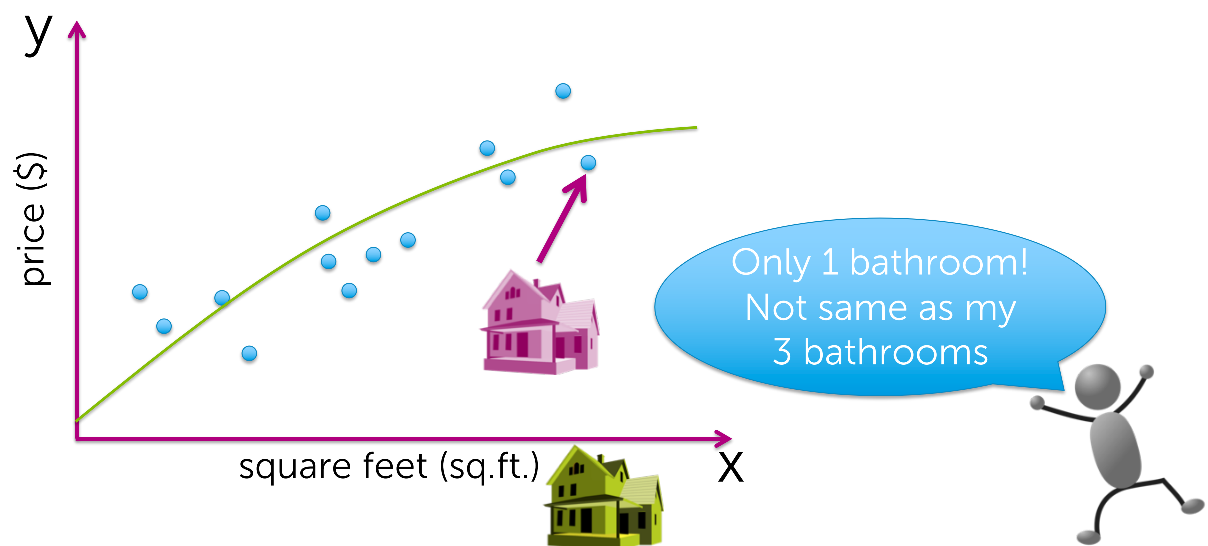
feature 3 = $h_2(x)$… e.g., x^2 or $\sin(2\pi x/12)$ or $\log(x)$

…

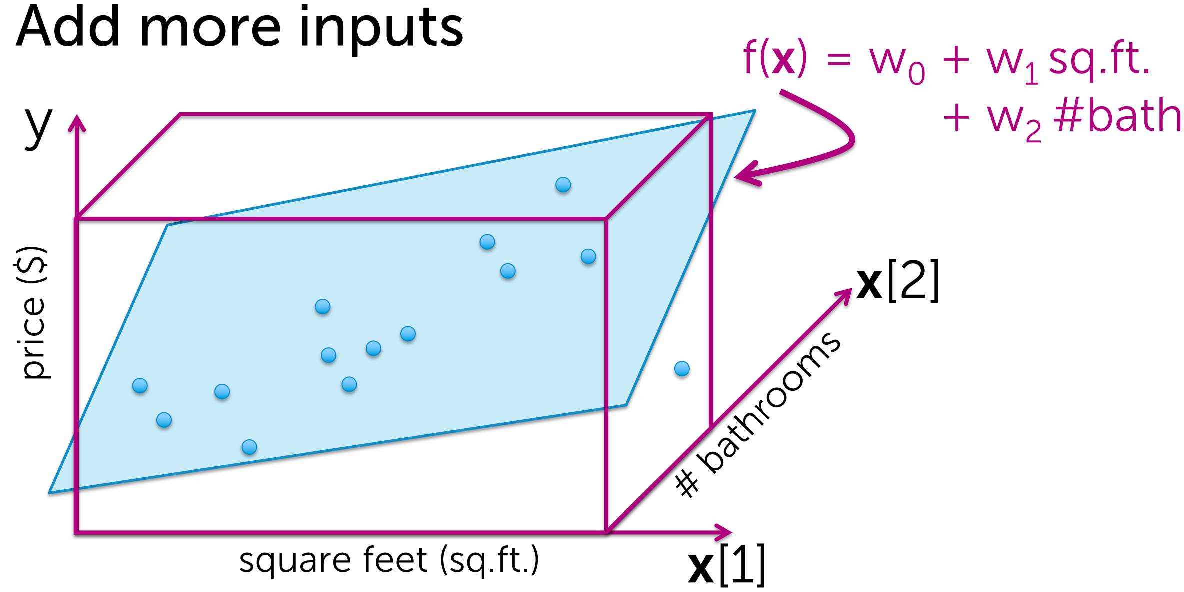
feature D+1 = $h_D(x)$… e.g., x^p

Adding other inputs

Predictions just based on house size



Add more inputs



Many possible inputs

- Square feet
- # bathrooms
- # bedrooms
- Lot size
- Year built
- ...

General notation

Output: y ← scalar

Inputs: $\mathbf{x}$ = ($\mathbf{x}$[1],$\mathbf{x}$[2],..., $\mathbf{x}$[d])

↖ d-dim vector

Notational conventions:

$\mathbf{x}$[j] = j^{th} input (*scalar*)

$h_j(\mathbf{x})$ = j^{th} feature (*scalar*)

$\mathbf{x}_i$ = input of i^{th} data point (*vector*)

$\mathbf{x}_i$[j] = j^{th} input of i^{th} data point (*scalar*)

Generic linear regression model

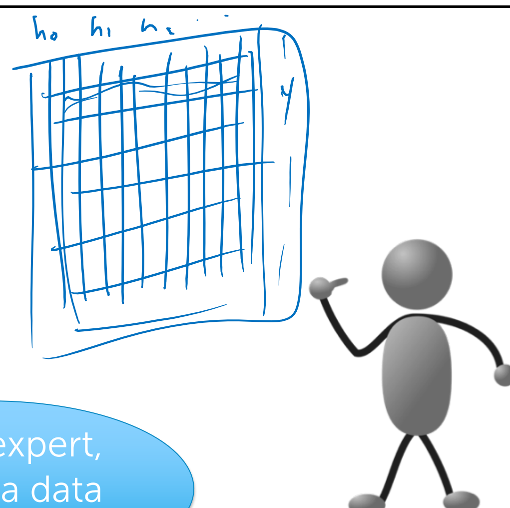
Model:

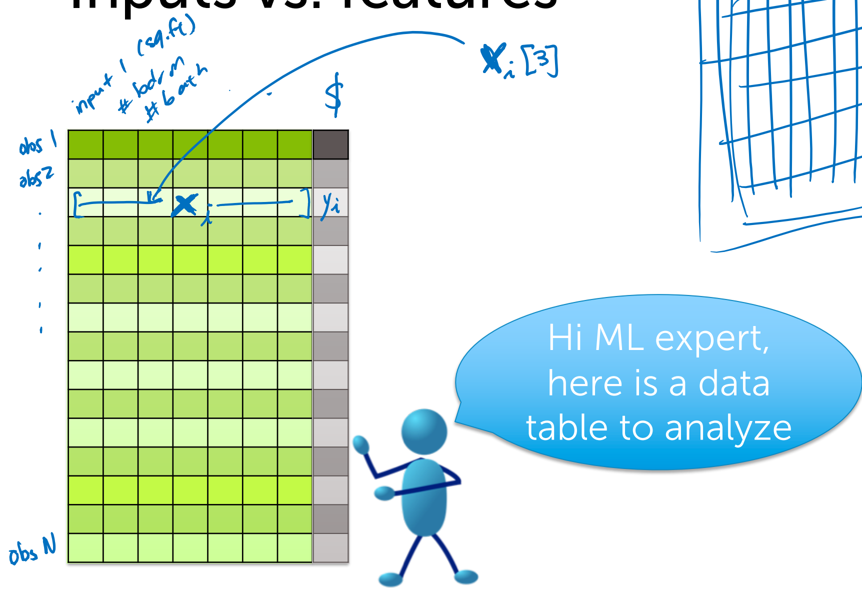
$$y_i = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i) + \varepsilon_i$$

$$= \sum_{j=0}^{D} w_j\, h_j(\mathbf{x}_i) + \varepsilon_i$$

feature 1 = $h_0(\mathbf{x})$ … e.g., 1
feature 2 = $h_1(\mathbf{x})$ … e.g., $\mathbf{x}[1]$ = sq. ft.
feature 3 = $h_2(\mathbf{x})$ … e.g., $\mathbf{x}[2]$ = #bath
or, $\log(\mathbf{x}[7])\ \mathbf{x}[2] = \log(\#\text{bed}) \times \#\text{bath}$
…
feature D+1 = $h_D(\mathbf{x})$ … some other function of $\mathbf{x}[1], \ldots, \mathbf{x}[d]$

Inputs vs. features





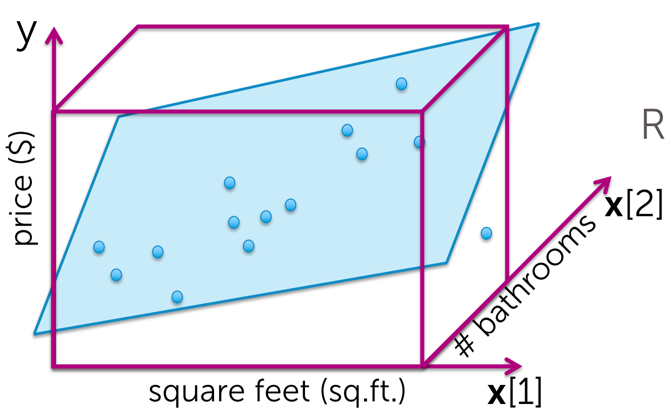
More on notation

observations $(\mathbf{x}_i, y_i)$: N

inputs $\mathbf{x}[j]$: d

features $h_j(\mathbf{x})$: D

RSS for multiple regression



$$RSS(\mathbf{w}) = \sum_{i=1}^{N} (y_i - \sum_{j=0}^{D} w_j h_j(\mathbf{x}_i))^2$$

actual obs

$f(\mathbf{x}_i)$

How many features to use?

- More on this soon!

A compact representation

Representing function using vectors

$$f(\mathbf{x}_i) = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i) = \sum_{j=0}^{D} w_j\, h_j(\mathbf{x}_i)$$

h: feat. 0, feat. 1, feat. 2 . . . feat. D

1	0	0	0	5	3	0	0	1	0	0	0	0

3	0	0	0	2	0	0	1	0	1	0	0	0

w: w_0 w_1 . . . w_D

$w_0 h_0 + w_1 h_1 + \ldots + w_D h_D$

$3 \cdot 1 + \ldots$

As an algorithm…

$$f(\mathbf{x}_i) = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i) = \sum_{j=0}^{D} w_j\, h_j(\mathbf{x}_i)$$

1	0	0	0	5	3	0	0	1	0	0	0	0

3	0	0	0	2	0	0	1	0	1	0	0	0

Algorithm:

for j=0,…,D

$f(\mathbf{x}_i) \mathrel{+}= w_j h_j(\mathbf{x}_i)$

Compact notation

$$f(\mathbf{x}_i) = w_0\, h_0(\mathbf{x}_i) + w_1\, h_1(\mathbf{x}_i) + \ldots + w_D\, h_D(\mathbf{x}_i) = \sum_{j=0}^{D} w_j\, h_j(\mathbf{x}_i)$$

1	0	0	0	5	3	0	0	1	0	0	0	0	$= h(x_i)$ feat. vec.
3	0	0	0	2	0	0	1	0	1	0	0	0	$= w$ weights vec.

write this operation as:

$w^T h(x_i)$

↑ notation