

- Join gradescope
 - there are instructions on piazza
- lots of info on website

Linear Regression: Model and Algorithms

CSE 446

Slides by Emily Fox (with minor modifications)

Presented by Anna Karlin

April 3, 2019

Linear regression: The model

How much is this house worth?



How much is this house worth?



Data



input *output*
 $(x_1 = \text{sq.ft.}, y_1 = \$)$



$(x_2 = \text{sq.ft.}, y_2 = \$)$



$(x_3 = \text{sq.ft.}, y_3 = \$)$



$(x_4 = \text{sq.ft.}, y_4 = \$)$



$(x_5 = \text{sq.ft.}, y_5 = \$)$

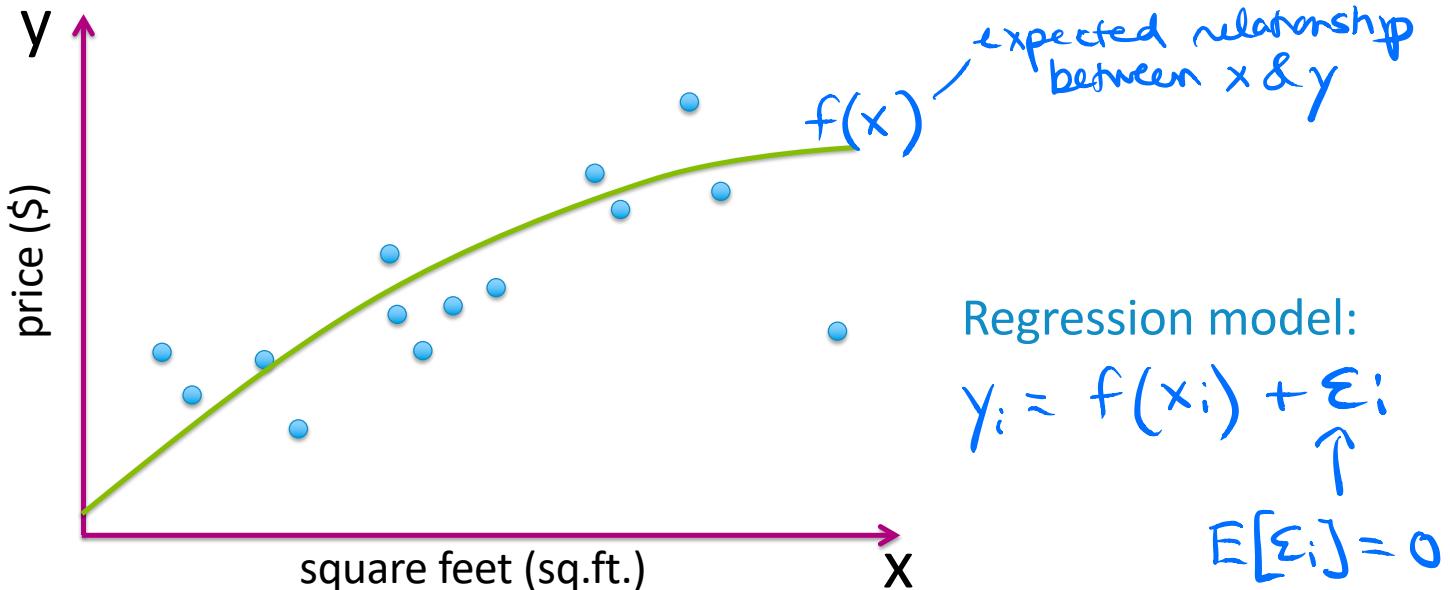
⋮

Input vs. Output:

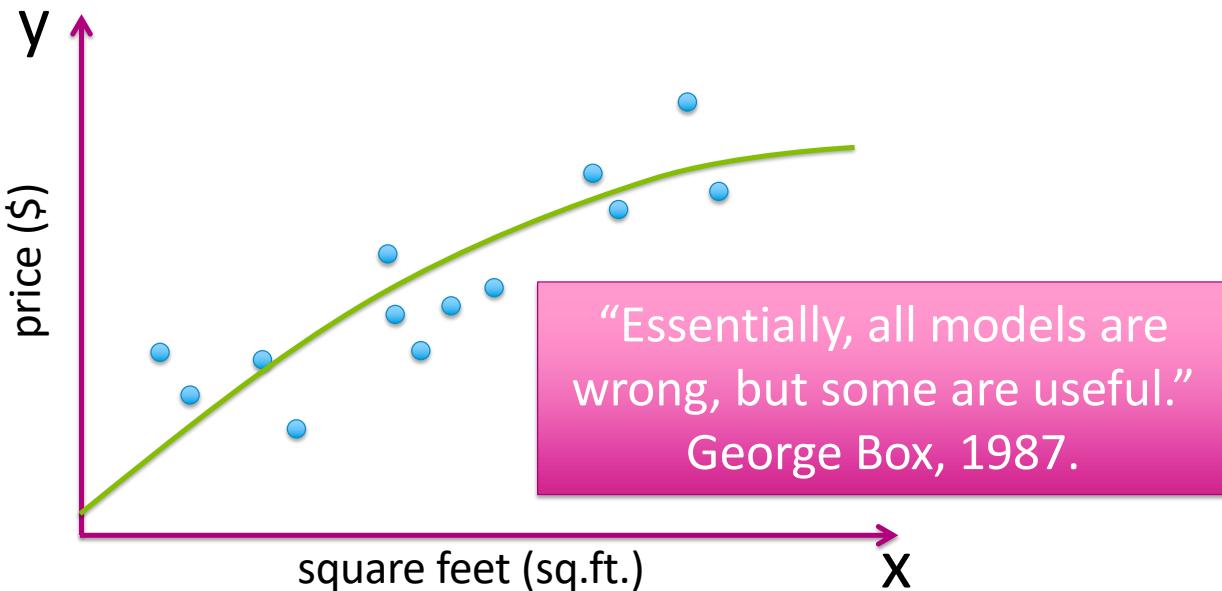
- y is the quantity of interest
- assume y can be predicted from x

Model –

How we *assume* the world works



Model – How we *assume* the world works



Process

- Chose a model

define a class of functions
 $\{f_w(x) \mid (w_1, \dots, w_n) \in \mathbb{R}^n\}$
s.t. we believe $\forall x \quad y = f_w(x) + \epsilon$

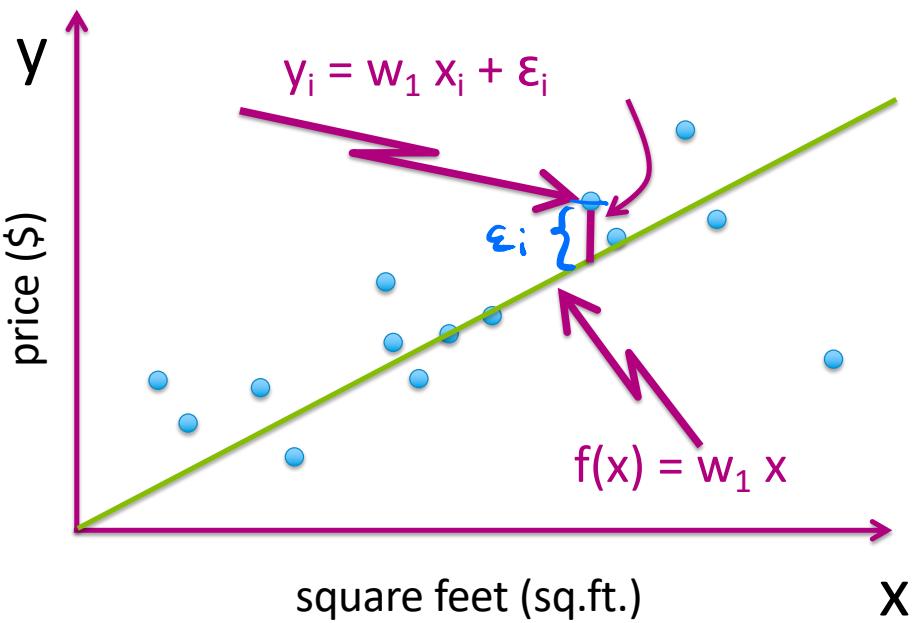
- Find the best fit to our data set.

what is best choice for parameters?
 \hat{w} best choice

- Use the fitted function to make future predictions.

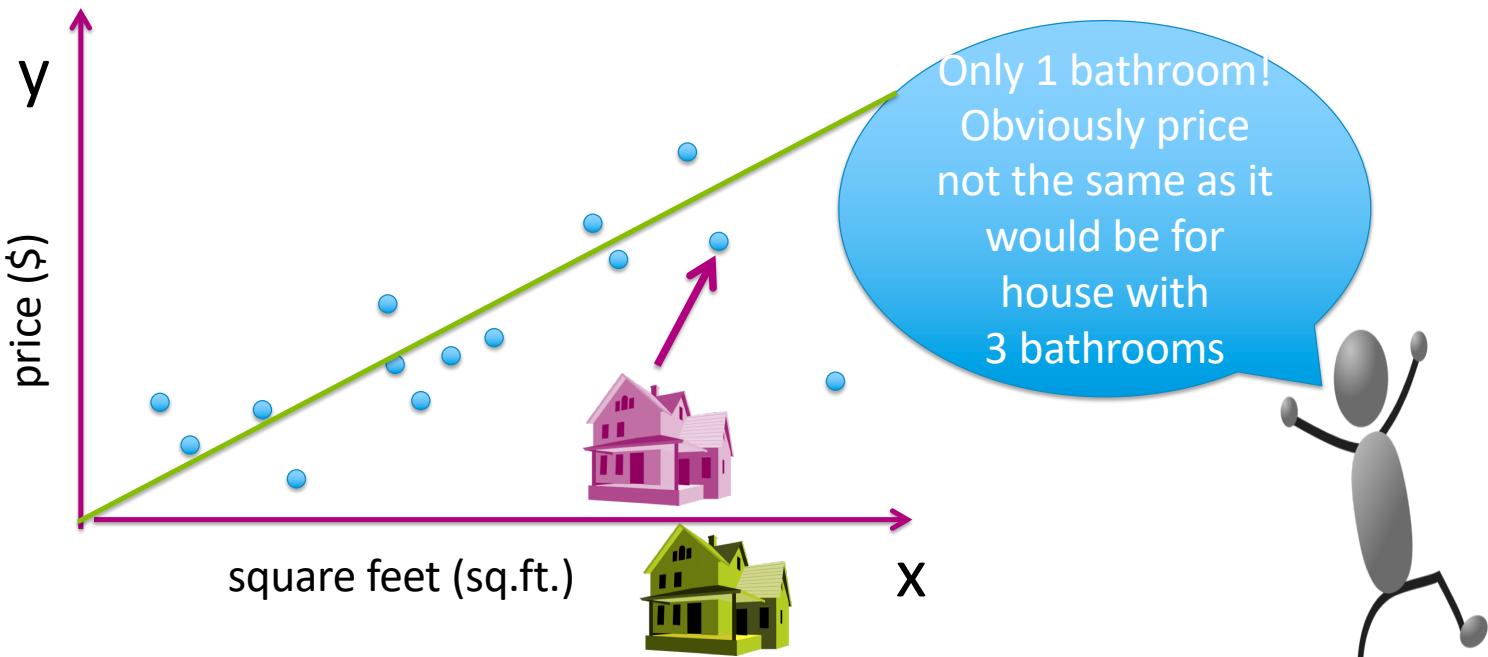
$$f_{\hat{w}}(x)$$

Simple linear regression model



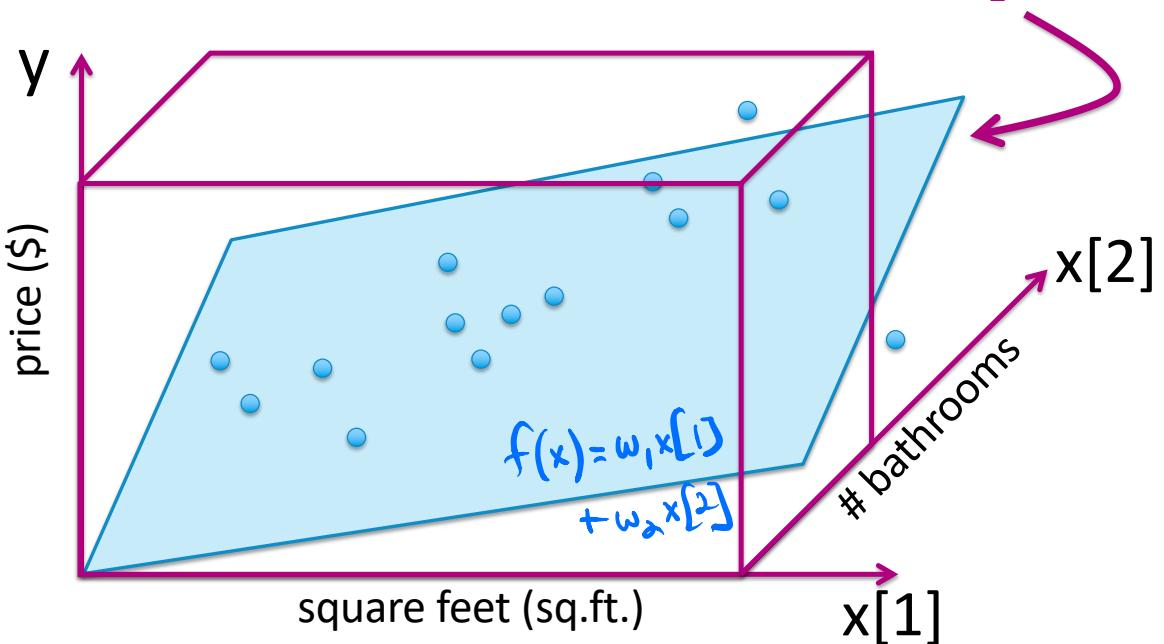
w_1 is the parameter
(regression coefficient)
That we want to learn
from our data set.

Predictions just based on house size



Add more inputs

$$f(x) = w_1 \text{sq.ft} + w_2 \# \text{bath}$$



Many possible inputs

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



A handwritten blue bracket is positioned above the first four items of the list, grouping them together. A blue arrow points from this bracket towards the word "features" written in blue cursive script to the right of the list.

features.

General notation

d features

Output: y *scalar*

Inputs: $x = (x[1], x[2], \dots, x[d])$ *d-dim vector*

e.g., $x[1] = \text{sq. ft}$, $x[2] = \#\text{baths}$ and so on.

Notational conventions:

training set: $\{(x_i, y_i)\}_{i=1..n}$

x_i = input of i^{th} data point/observation (*vector*); y_i is output

$x_i[j]$ = j^{th} input of i^{th} data point (*scalar*)

n = number of observations; d = number of input features

Generic linear regression model

Model: Given feature vector $\mathbf{x}_i = (x_i[1], x_i[2], \dots, x_i[d])$

$$y_i = w_1 x_i[1] + w_2 x_i[2] + \dots + w_d x_i[d] + \varepsilon_i$$

\equiv

$$= \sum_{j=1}^d w_j x_i[j] + \varepsilon_i \quad = \mathbf{w}^T \mathbf{x}_i + \varepsilon_i = \mathbf{x}_i^T \mathbf{w} + \varepsilon_i$$

Goal: find best choice for params w_1, \dots, w_d

feature 1 = $x[1]$ = sq. ft.

feature 2 = $x[2]$ = #bath

...

feature $d = x[d]$ = lot size

$$\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}$$

$$(w_1 \dots w_d) \begin{pmatrix} x_i[1] \\ \vdots \\ x_i[d] \end{pmatrix}$$

Fitting the linear regression model

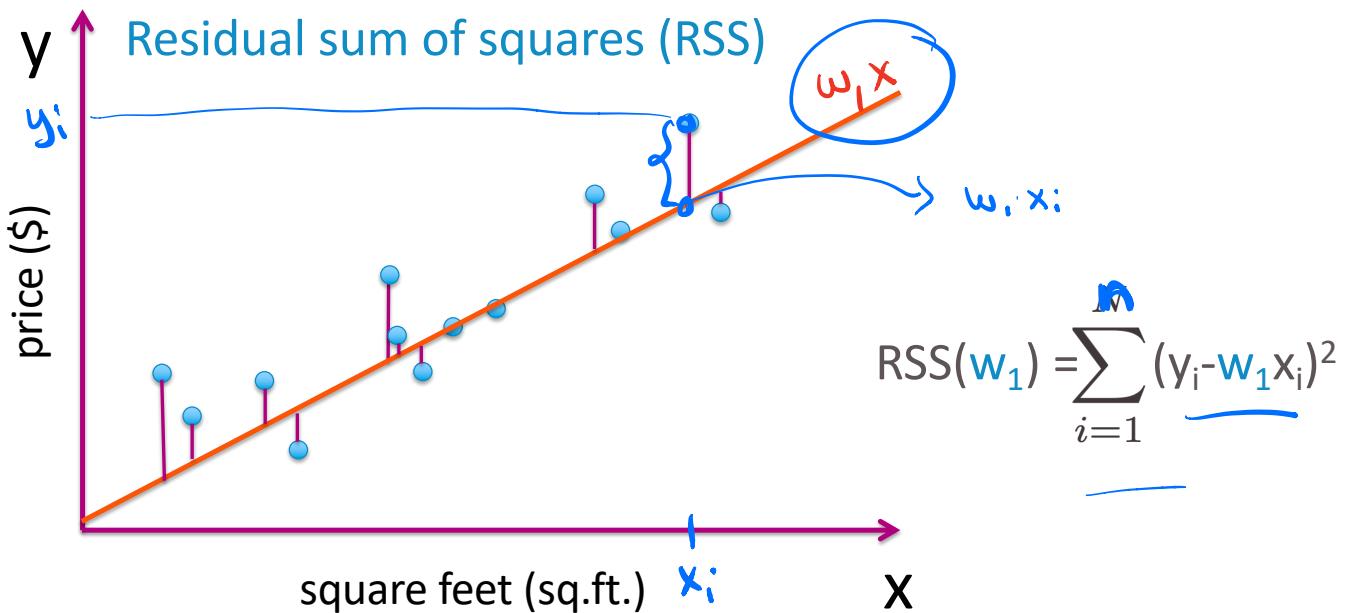
- model
- training set

find best $\vec{\omega}$

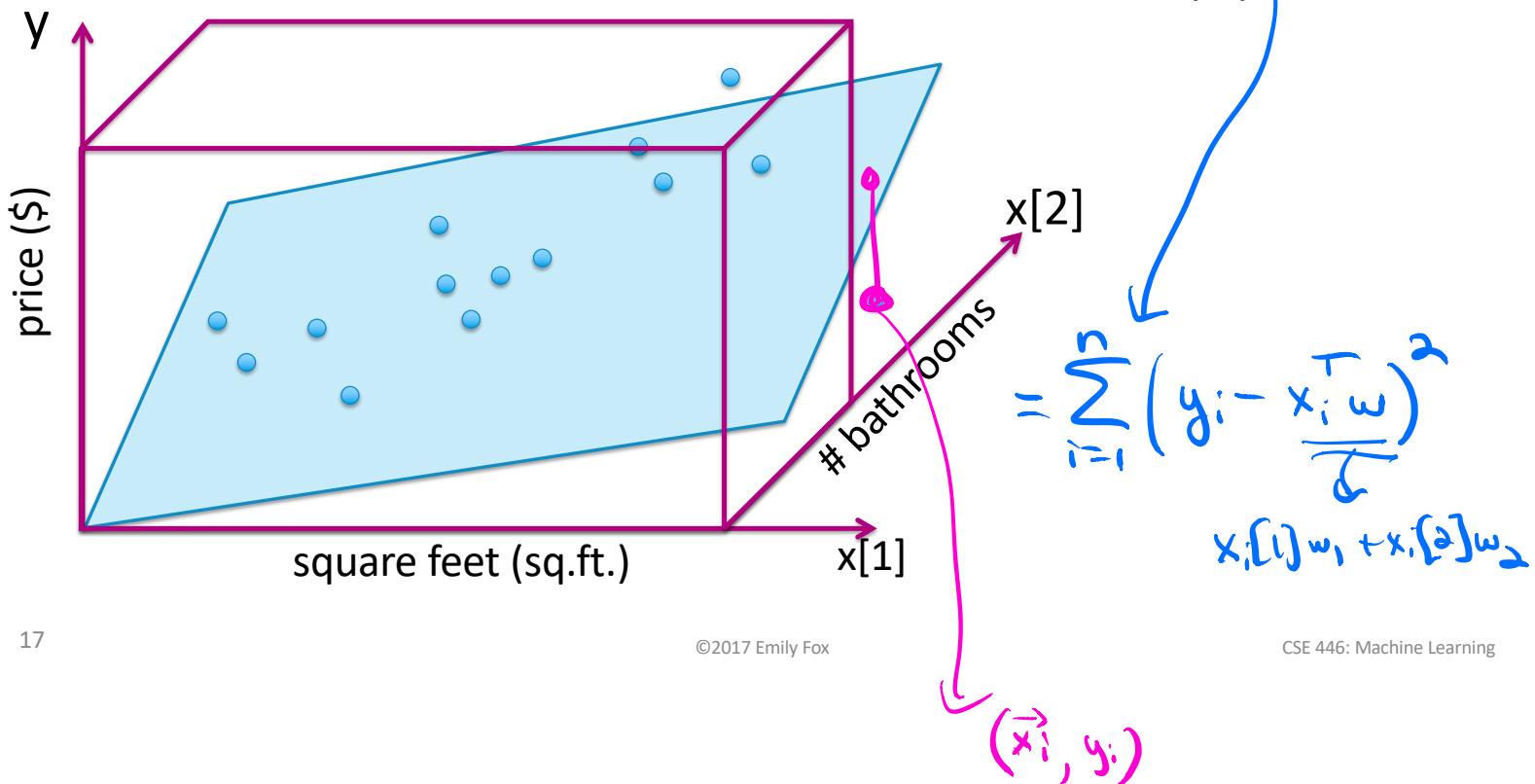
fit of parameter set $\vec{\omega}$
to training set.

cost ($\vec{\omega}$)
loss ($\vec{\omega}$)

“Cost” of using a given line



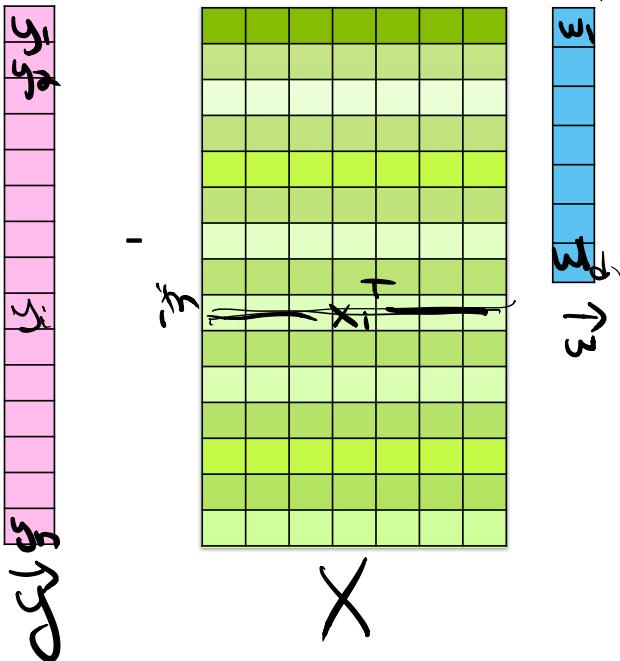
RSS for multiple regression



$$\text{RSS}(\omega) = \sum_{i=1}^n (y_i - x_i^\top \omega)^2 = (y - X\omega)^\top (y - X\omega) \\ = \|y - X\omega\|^2$$

Rewrite in matrix notation

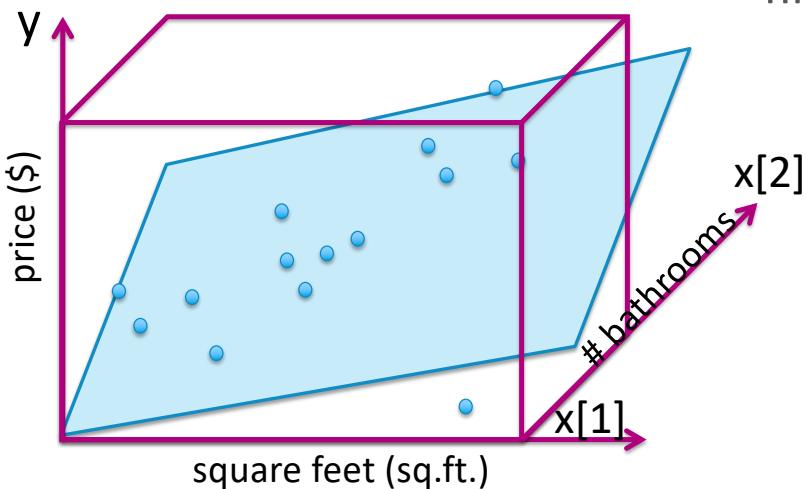
For all observations together



$$\vec{y} - \vec{X}\vec{w} = \begin{pmatrix} y_1 - x_1^\top w \\ y_2 - x_2^\top w \\ \vdots \\ y_n - x_n^\top w \end{pmatrix}$$



RSS for multiple regression



Objective: Find the best fit, i.e., find the \mathbf{w} that minimizes

$$\text{RSS}(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$w_1, \dots, w_d$$

Interlude: Optimization (and convex functions)

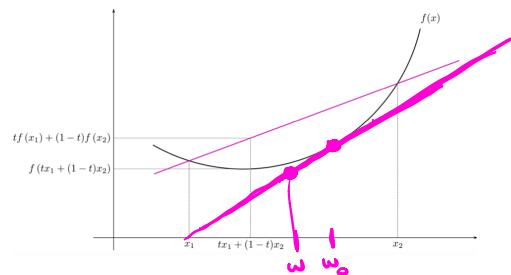
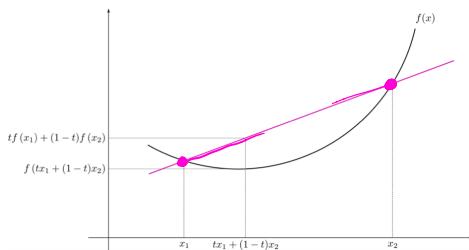
Minimizing a univariate function

$$f(w) \quad \text{to} \quad f'(w)=0 \\ \Rightarrow f''(w)>0$$

for convex fns $f''(w)=0 \Rightarrow w$ is global minimum

Convex functions \Leftrightarrow ① $f''(w) \geq 0$ holds if w

② $\forall w$ tangent line is global underestimator.



$l(w) = f(w_0) + (w-w_0)f'(w_0) \leq f(w)$

graph
tang
line
that is tangent at w_0

another
defn for what it
means for a fn to be
convex.

$$f(w_1, \dots, w_d)$$

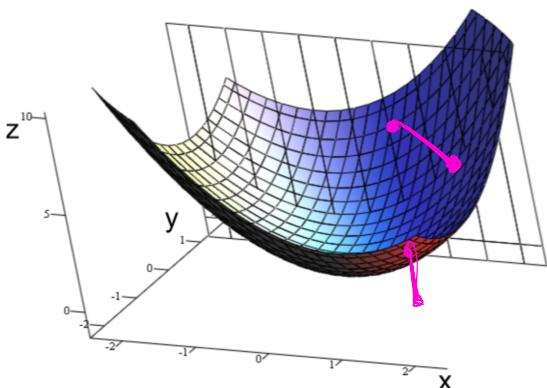
Minimizing a multivariate function

gradient.

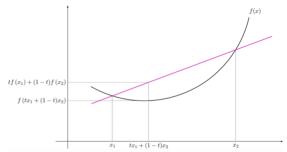
$$\nabla f(\vec{w}) = \begin{pmatrix} \vdots \\ \frac{\partial f}{\partial w_i}(\vec{w}) \\ \vdots \end{pmatrix} \rightarrow f'(w) = \nabla f(w) = \vec{0}$$

$$\nabla f(\omega) = 0 \Rightarrow \text{global min}$$

Convex functions

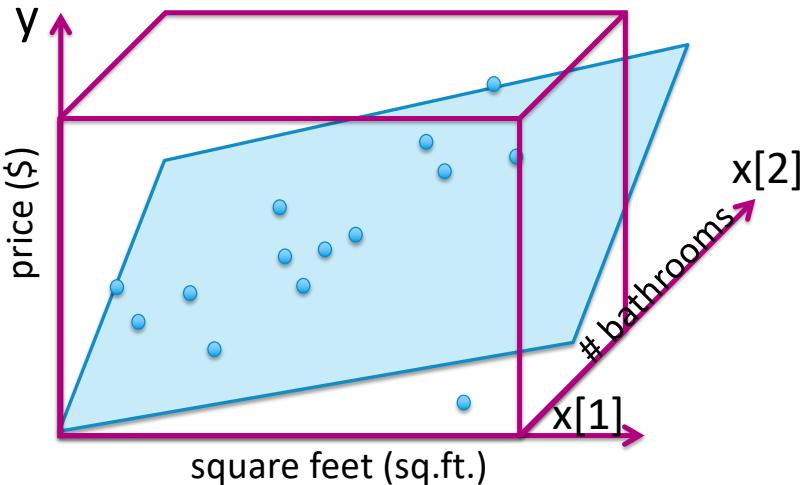


$$f(\vec{w}_0) + (\vec{w} - \vec{w}_0)^T \nabla f(\vec{w}_0) \leq f(\vec{w}) \quad \forall \vec{w}.$$



Back to our specific optimization problem

Two ways to solve our optimization problem



Objective: Find the best fit, i.e., find the w that minimizes

$$\text{RSS}(w) = (y - Xw)^T(y - Xw)$$

RSS (w) is a convex function of w

1. Solve for $\nabla \text{RSS}(\mathbf{w}) = 0$

Gradient of RSS

$$\vec{\omega} = \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_d \end{pmatrix}$$

$$\nabla \text{RSS}(\omega) = \nabla[(y - X\omega)^T(y - X\omega)]$$

$$\text{RSS}(\omega) = \sum_i (y_i - x_i^T \omega)^2$$

$$\Rightarrow \frac{\partial \text{RSS}(\omega)}{\partial \omega_j} \doteq \sum_i 2(y_i - x_i^T \omega) \leftarrow x_i[j]$$

$$\nabla \text{RSS}(\omega) = -2X^T(y - X\omega)$$

$$X = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \end{pmatrix}$$

$$x_i^T \omega = \sum_{l=1}^d x_{i,l} \omega_l$$

$\leftarrow l=1$

$$-2 \begin{bmatrix} | & | & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & \cdots & | \end{bmatrix} \begin{pmatrix} | \\ y_i - x_i^T \omega \\ | \end{pmatrix}$$

X^T

$y - X\omega$

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix}$$

Closed form solution

$$\nabla \text{RSS}(w) = \nabla[(y - Xw)^T(y - Xw)] \\ = -2X^T(y - Xw)$$

We want solution to $-2X^T(y - X\hat{w}) = 0$

$$X^T y - X^T X \hat{w} = 0$$

w where
 $\nabla \text{RSS} = 0$.

Solution: "normal equations"

$$X^T X \hat{w} = X^T y$$

if $X^T X$
is invertible

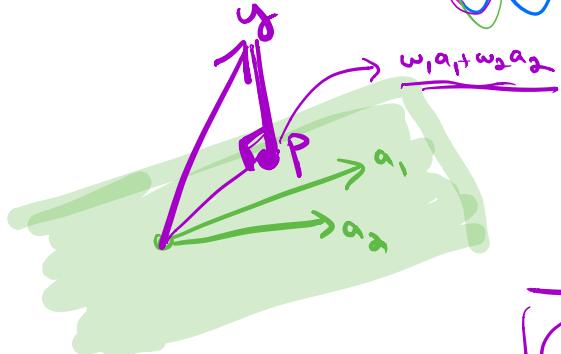
$$\hat{w} = (X^T X)^{-1} X^T y$$

dxd.

$O(d^3)$

Geometric intuition for solution

Suppose $X = \begin{pmatrix} a_1 & a_2 \\ 1 & 0 \\ 1 & 1 \\ 1 & 3 \end{pmatrix}$ looking for RSS



$$y - p \perp \text{plane}$$

$$p = Xw$$

w that minimizes $\|y - Xw\|^2$

~~$y = Xw$~~

$$= w_1 \cdot \vec{a}_1 + w_2 \cdot \vec{a}_2$$

$$\begin{aligned} a_1^T (y - Xw) &= 0 \\ a_2^T (y - Xw) &= 0 \end{aligned}$$

$$X^T (y - \hat{Xw}) = 0$$

$$\begin{pmatrix} a_1^T \\ a_2^T \end{pmatrix}$$

\hat{Xw} : predicted sales prices

Again, back to getting closed form solution.

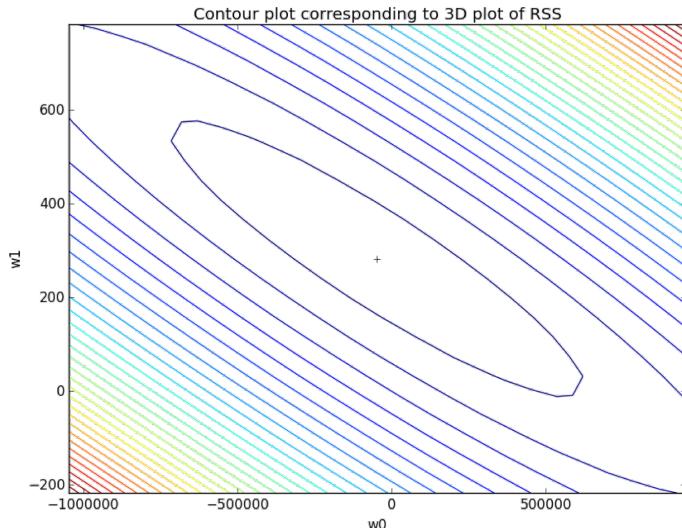
2. Gradient descent

Gradient Descent – univariate case

- Repeatedly move in direction that reduces the value of the function.

Gradient Descent – multivariate case

Gradient descent for linear regression: repeatedly move in direction of negative gradient

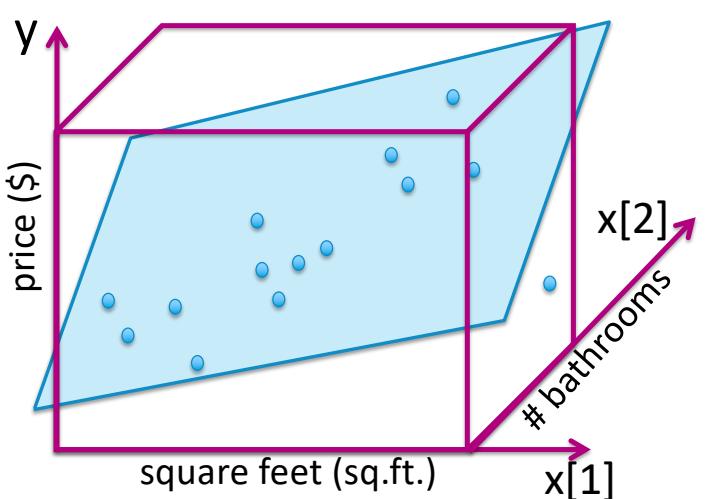


while not converged

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

$\underbrace{-2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}^{(t)})}$

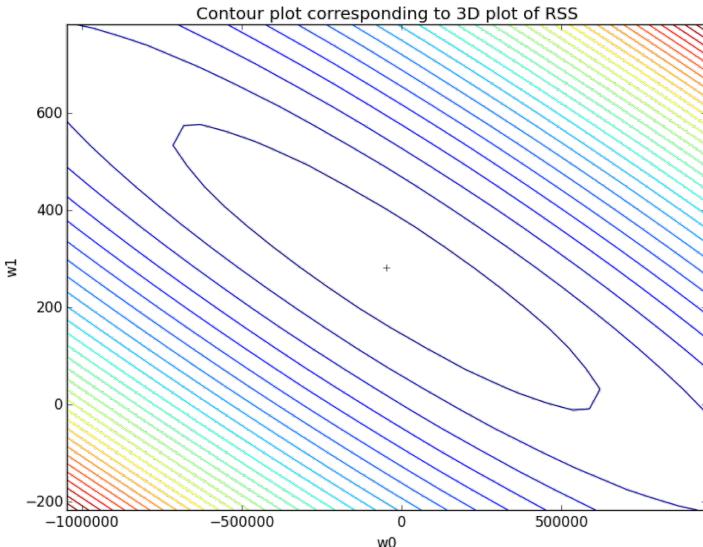
Interpreting elementwise



Update to j^{th} feature weight:

$$w_j^{(t+1)} \leftarrow w_j^{(t)} + 2n \sum_{i=1}^N x_i[j](y_i - \hat{y}_i(w^{(t)}))$$

Summary of gradient descent for multiple regression



```
init  $w^{(1)}=0$  (or randomly, or smartly),  $t=1$ 
while  $\|\nabla \text{RSS}(w^{(t)})\| > \varepsilon$ 
    for  $j=1,\dots,d$ 
         $\text{partial}[j] = -2 \sum_{i=1}^n x_i[j](y_i - \hat{y}_i(w^{(t)}))$ 
         $w_j^{(t+1)} \leftarrow w_j^{(t)} - \eta \text{partial}[j]$ 
     $t \leftarrow t + 1$ 
```

Adding an intercept – “demeaning”

Once we have a fitted function

- We use it to predict the sales price for new houses, by plugging in square footage, number of bathrooms, etc for the new house \mathbf{x} whose sales price we want to predict.
- Prediction is:
- What if we want to allow for an intercept?

Handling an intercept (constant term)

Two step approach:

1. Show that if $\frac{1}{n} \sum_i \mathbf{x}_i = \mathbf{0}$ (*) then solution is simple.
2. Show how to transform, aka ``demean'' any linear regression problem so that (*) holds.

1. Show that if $\frac{1}{n} \sum_i x_i = 0$ (*) then solution is simple.

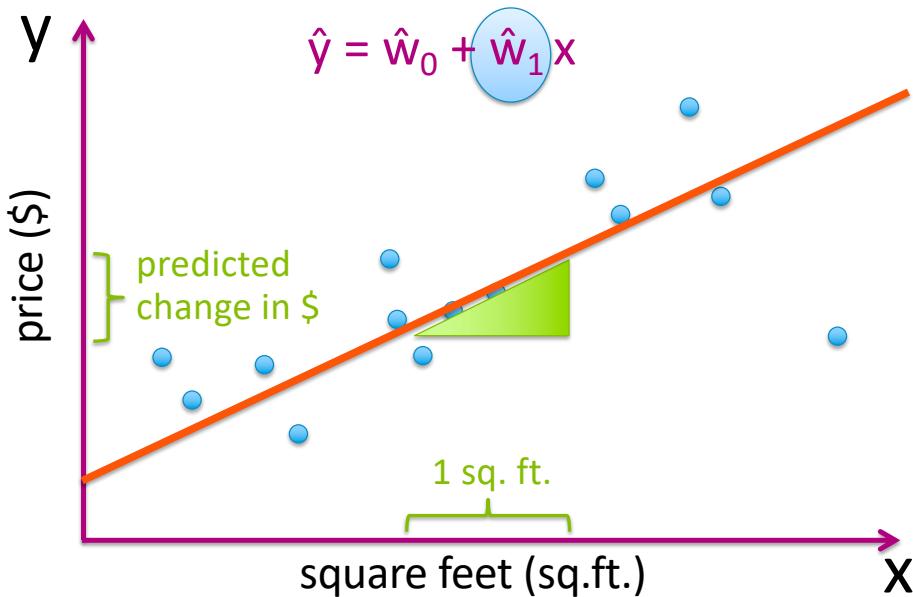
Same as saying that $X^\top \mathbf{1} = 0$.

2. Show how to transform, aka ``demean'' any linear regression problem so that (*) holds.

$$\frac{1}{n} \sum_i \mathbf{x}_i = \mathbf{0} \quad (*)$$

Interpreting the fitted function

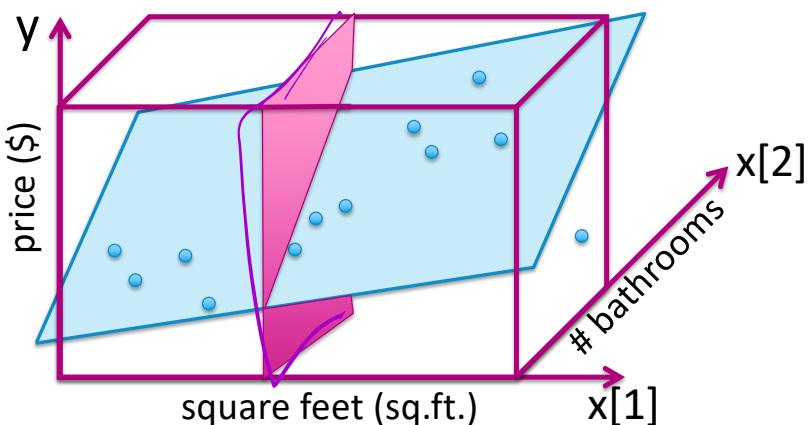
Interpreting the coefficients – Simple linear regression



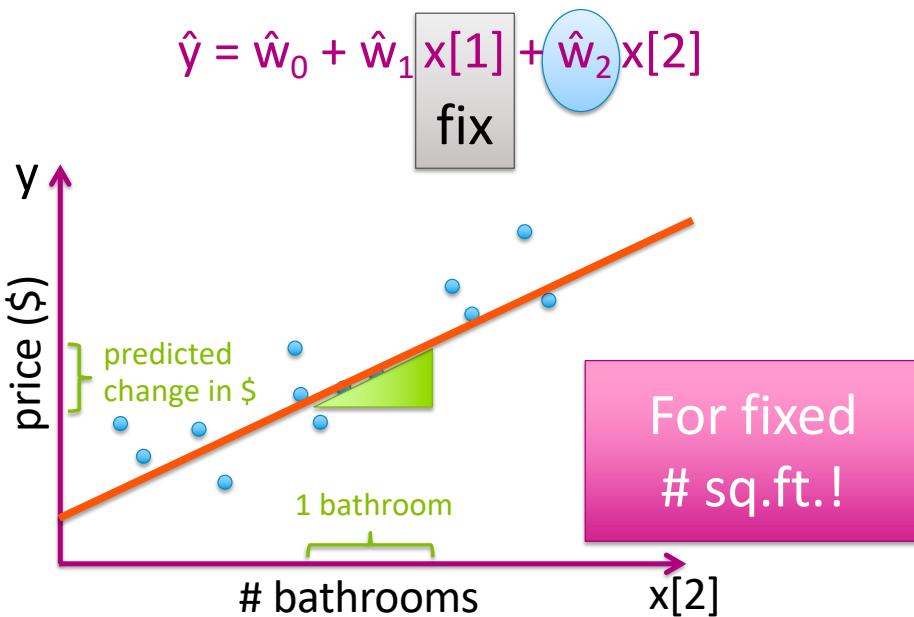
Interpreting the coefficients – Two linear features

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x[1] + \hat{w}_2 x[2]$$

fix



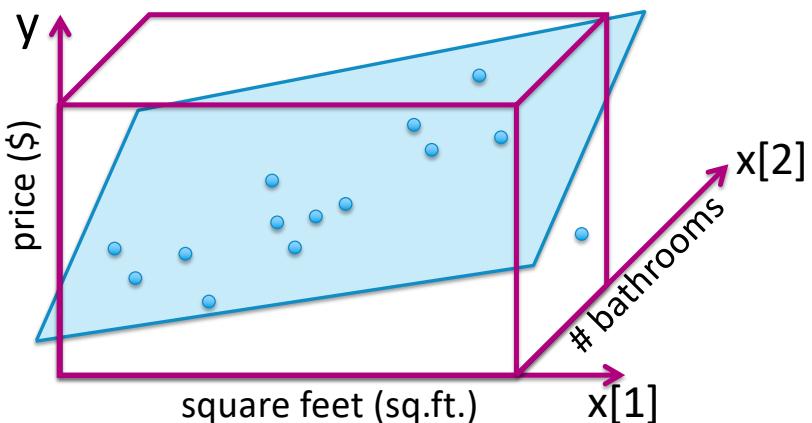
Interpreting the coefficients – Two linear features



Interpreting the coefficients – Multiple linear features

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x[1] + \dots + \hat{w}_j x[j] + \dots + \hat{w}_d x[d]$$

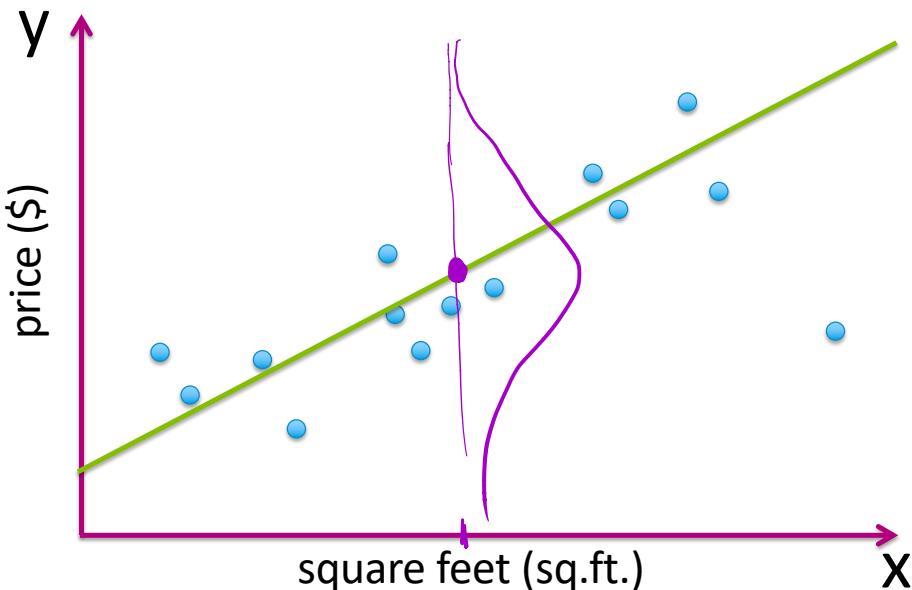
fix fix



Why min RSS?

Assuming Gaussian noise

$$y_i = \underline{\underline{w^T x_i}} + \epsilon_i$$



Model for ϵ_i :

$$N(0, \sigma^2)$$

Implied distribution
on y_i :

$$N(w^T x_i, \sigma^2)$$

Maximum likelihood estimate of params

If $\underline{y}_i = \mathbf{x}_i^T \mathbf{w} + \varepsilon_i$ is $N(\mathbf{x}^T \mathbf{w}, \sigma^2)$

And we see data $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, what is the MLE of \mathbf{w} ?

$$\log P(\underline{y}_1, \dots, \underline{y}_n | \mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{w}, \sigma^2) = \log \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_i - \mathbf{x}_i^T \mathbf{w})^2}{2\sigma^2}}$$

$$\log P = \sum_{i=1}^n \left[\log \left(\frac{1}{\sqrt{2\pi}\sigma} \right) - \frac{(y_i - \mathbf{x}_i^T \mathbf{w})^2}{2\sigma^2} \right]$$

$$\max \text{ log likelihood} \equiv \max \sum_{i=1}^n -\frac{(y_i - \mathbf{x}_i^T \mathbf{w})^2}{2\sigma^2} \equiv \min \sum_{i=1}^n (y_i - \mathbf{x}_i^T \mathbf{w})^2$$

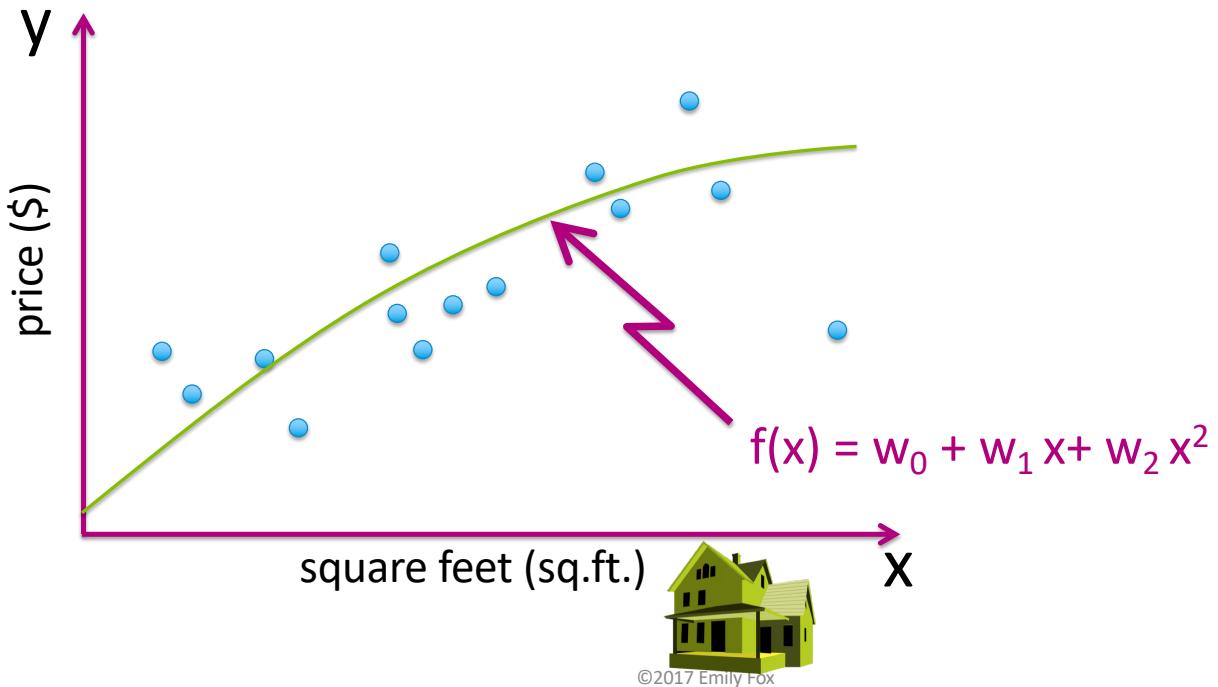
Conclusion

If $y_i = x_i^T \mathbf{w} + \varepsilon_i$ is $N(x^T \mathbf{w}, \sigma^2)$

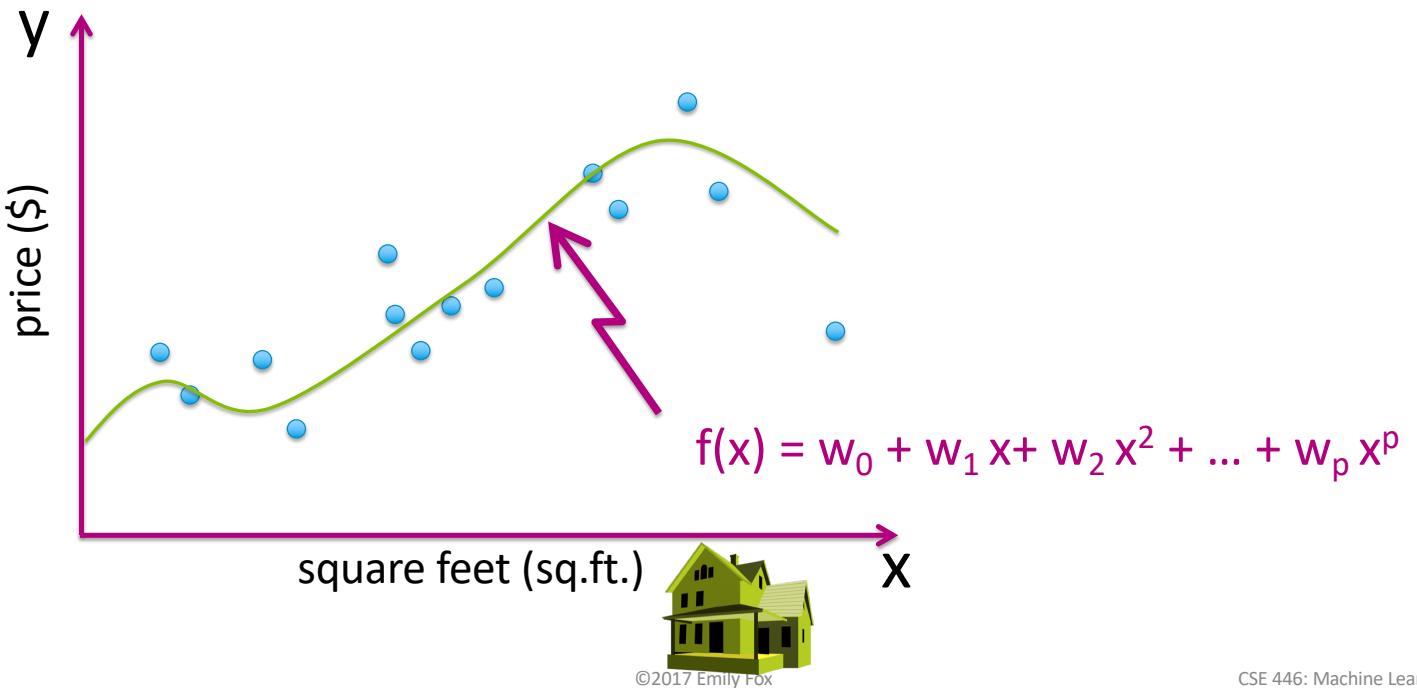
Then choosing \mathbf{w} to maximize log-likelihood
is same as choosing \mathbf{w} to minimize RSS!

Feature maps, polynomial regression and basis expansion

What about a quadratic function?



Even higher order polynomial



Polynomial Regression

- Start with single input feature x (e.g. square footage of house) and training set: $\{(x_i, y_i)\}_{i=1..n}$
- Define feature map that transforms each x_i to higher dimensional feature vector $h(x_i)$.

Example: x_i

$$\begin{array}{ccccccc} h_0(x_i) & h_1(x_i) & h_2(x_i) & h_3(x_i) & h_4(x_i) & h_5(x_i) \\ 1 & x_i & x_i^2 & x_i^3 & x_i^4 & x_i^5 \end{array}$$

Polynomial regression

Model:

$$y_i = w_0 + w_1 x_i + w_2 x_i^2 + \dots + w_p x_i^p + \epsilon_i$$

treat transformed inputs as different features

feature 1 = 1 (constant)

feature 2 = x

feature 3 = x²

...

feature p+1 = x^p

parameter 1 = w₀

parameter 2 = w₁

parameter 3 = w₂

...

parameter p+1 = w_p

Why might we want to use polynomial regression?

- Taylor Series!

More generally

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

Example: $x_i[1] \ x_i[2] \ x_i[3]$

$$\begin{array}{ccccccc} h_1(\mathbf{x}) & h_2(\mathbf{x}_i) & h_3(\mathbf{x}_i) & h_4(\mathbf{x}_i), & h_5(\mathbf{x}_i) & h_6(\mathbf{x}_i) & h_7(\mathbf{x}_i) \\ 1 & x_i[1] & x_i[1]^2 & x_i[1]x_i[2] & x_i[2] & x_i[2]^2 & \cos(\pi x_i[3]/6) \end{array}$$

General notation

Output: y  scalar

Inputs: $x = (x[1], x[2], \dots, x[d])$  d-dim vector

Notational conventions:

x_i = input of i^{th} data point (*vector*)

$x_i[j]$ = j^{th} input of i^{th} data point (*scalar*)

$h(x) = (h_1(x), h_2(x), \dots, h_d(x))$ feature map applied to input x (*vector*)

$h_j(x)$ = j^{th} feature associated with input x (*scalar*) (j^{th} basis function)

To fit these more general functions

- Start with input features $\mathbf{x} = (x[1], x[2], \dots, x[d])$ 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 $h(\mathbf{x}_i)$.
- Model: $y_i = \sum_{j=1}^p w_j h_j(\mathbf{x}_i) + \varepsilon_i$
- Find $\hat{\mathbf{w}}$ that minimizes $\text{RSS} = \sum_{i=1}^n (y_i - \sum_{j=1}^p w_j h_j(\mathbf{x}_i))^2$

Recap of concepts

What you can do now...

- Describe linear regression (and feature maps)
- Write a regression model using multiple inputs or features thereof.
- Calculate a goodness-of-fit metric (e.g., RSS)
- Estimate model parameters of a general multiple regression model to minimize RSS:
 - In closed form
 - Using an iterative gradient descent algorithm
- Interpret the coefficients of a non-featurized multiple regression fit
- Exploit the estimated model to form predictions