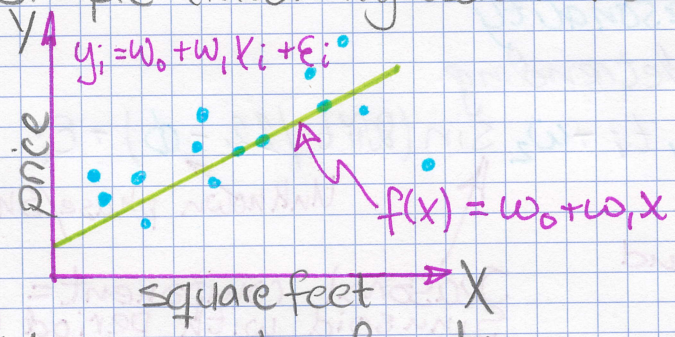
Multiple Regression

Linear regression with multiple features

Polynomial regression

Simple linear regression model

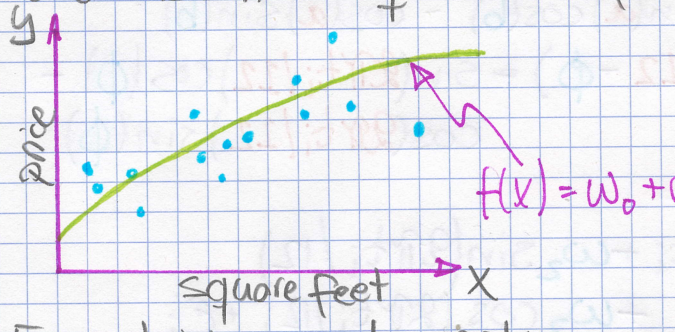


$y_i = w_0 + w_1 x_i + \varepsilon_i$

$f(x) = w_0 + w_1 x$

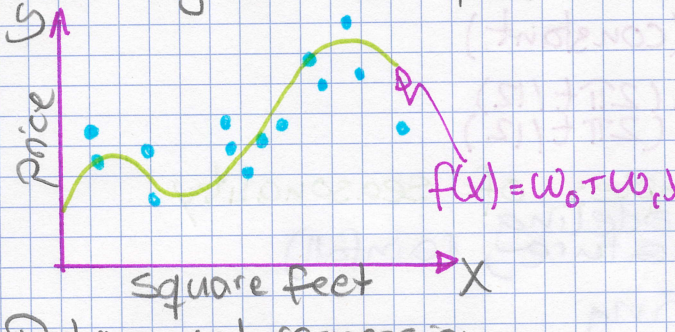
More complex functions of a single input

What about a quadratic function?



$f(x) = w_0 + w_1 x + w_2 x^2$

Ever higher order polynomial



$f(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_p x^p$

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)
feature 2 = x
feature 3 = x^2
...
feature $p+1 = x^p$

parameter 1 = ω_0
parameter 2 = ω_1
parameter 3 = ω_2
...
parameter $p+1 = \omega_p$

Modeling seasonality

An example detrending

$$y_i = \omega_0 + \omega_1 t_i + \omega_2 \sin(2\pi t_i/12 - \phi) + \epsilon_i$$

$\omega_1 t_i$: Linear trend

$\omega_2 \sin(2\pi t_i/12 - \phi)$: Seasonal component = Sinusoid with period 12 (resets annually)

ϕ: Unknown phase/shift

Trigonometric identity:

$$\sin(a-b) = \sin(a)\cos(b) - \cos(a)\sin(b)$$

$$\rightarrow \sin(2\pi t_i/12 - \phi) = \sin(2\pi t_i/12)\cos(\phi) - \cos(2\pi t_i/12)\sin(\phi)$$

Equivalent

$$y_i = \omega_0 + \omega_1 t_i + \omega_2 \sin(2\pi t_i/12) + \omega_3 \cos(2\pi t_i/12) + \epsilon_i$$

feature 1 = 1 (constant)
feature 2 = t
feature 3 = $\sin(2\pi t/12)$
feature 4 = $\cos(2\pi t/12)$

Other examples of seasonality

Weather modeling (e.g. temperature, rainfall)
Flu monitoring
Demand forecasting (e.g. jacket purchases)
Motion capture data

Regression with general features of 1 input

Generic basis expansion

Model:

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

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

w_j: j^{th} regression coefficient or weight

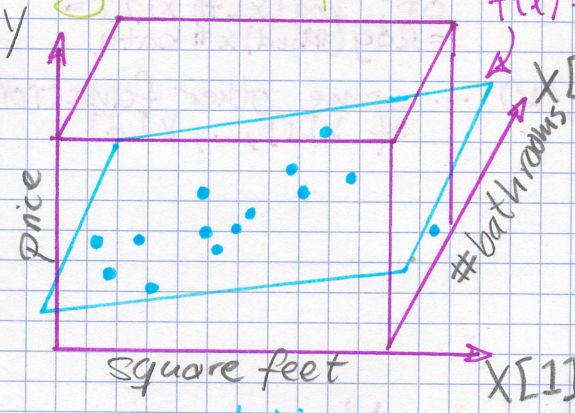
$h_j(x_i)$: j^{th} feature

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 $\sin(2\pi x/12)$
…
feature D+1 = $h_D(x)$ … e.g., x^p

Motivating the use of multiple inputs

Adding more inputs

$f(x) = w_0 + w_1 \text{sq.ft} + w_2 \#\text{bath}$



Many possible inputs

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

General notation

Output: y ← scalar

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

Notational conventions:

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

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

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

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

Regression with features of multiple inputs

Simple hyperplane

Model:

$$y_i = w_0 + w_1 x_i[1] + \ldots + w_d x_i[d] + \varepsilon_i$$

feature 1 = 1

feature 2 = x[1] sq.ft

feature 3 = x[2] #bath

...

feature d+1 = x[d] lot size

More generically D-dimensional curve

Model:

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

feature 1 = $h_0(x)$ ... e.g. 1

feature 2 = $h_1(x)$ ... e.g. x[1] = sq.ft

feature 3 = $h_2(x)$ ... e.g. x[2] = #bath

or, log(x[7]) x[2] = log(#bed) x #bath

...

feature D+1 = $h_D(x)$ ... some other function of x[1], ..., x[d]

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

More on notation

observations (x_i, y_i) : N

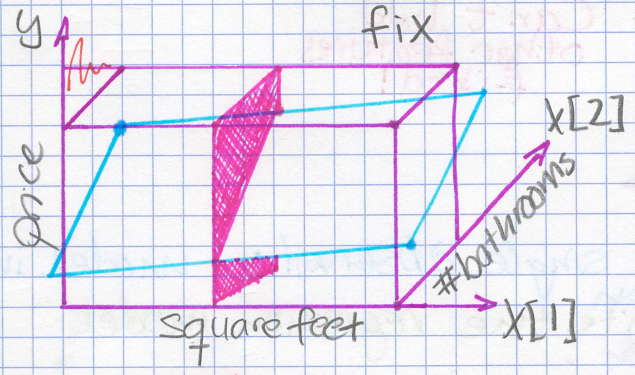
inputs x[j] : d

features $h_j(x)$: D

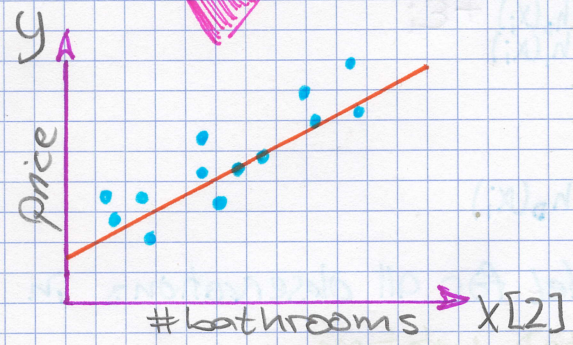
Interpreting the multiple regression fit

Interpreting the coefficients - Two linear features

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



For fixed #Sq.ft

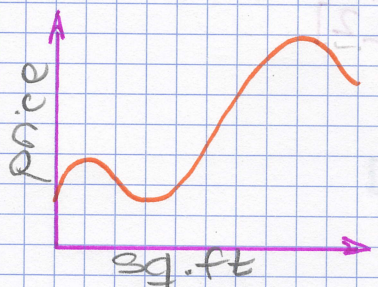


Interpreting the coefficients - Multiple linear features

$$\hat{y} = \hat{w}_0 + \hat{w}_1 \underbrace{x[1]}_{\text{fix}} + \underbrace{\ldots}_{\text{fix}} + \hat{w}_j x[j] + \underbrace{\ldots}_{\text{fix}} + \hat{w}_d \underbrace{x[d]}_{\text{fix}}$$

Interpreting the coefficients - Polynomial regression

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x + \ldots + \hat{w}_j x^j + \ldots + \hat{w}_p x^p$$



Can't hold other features fixed!

Rewriting the single observation model in vector notation

Step 1: Rewrite the regression model

Matrix notation

For observation i

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

$$y_i = \underbrace{\begin{bmatrix} w_0 & w_1 & \cdots & w_D \end{bmatrix}}_{w^T} \overbrace{\begin{bmatrix} h_0(x_i) \\ h_1(x_i) \\ \vdots \\ h_D(x_i) \end{bmatrix}}^{h(x_i)} + \varepsilon_i$$

$$= w^T h(x_i) + \varepsilon_i$$

Rewriting the model for all observations in matrix notation

Rewrite in matrix notation

For all observations together

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} h_0(x_1) & h_1(x_1) & \cdots & h_D(x_1) \\ h_0(x_2) & & & h_D(x_2) \\ & & H & \\ h_0(x_N) & & & h_D(x_N) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_D \end{bmatrix}$$

$h_0(x_1)$ $h_1(x_1)$ $\cdots$ $h_D(x_1)$ $= h^T(x_1)$

$$\Rightarrow \boxed{Y = Hw + \varepsilon}$$

Computing the cost of a D-dimensional curve

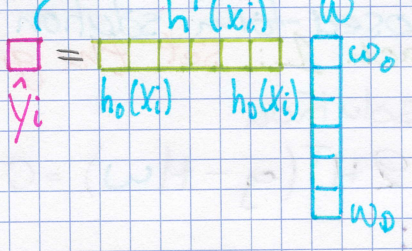
Step 2: Compute the cost

$$RSS(w_0, w_1) = \sum_{i=1}^{N} \left(y_i - \underbrace{[w_0 + w_1 x_i]}_{\hat{y}_i(w_0, w_1)}\right)^2$$

RSS for multiple regression

$$RSS(w) = \sum_{i=1}^{N} \left(y_i - h^T(x_i) w\right)^2$$

$h^T(x_i)$ produces $\hat{y}_i(w)$



RSS in matrix notation $RSS(w) = \sum_{i=1}^{N} \left(y_i - h(x_i)^T w\right)^2$

$$= (y - Hw)^T (y - Hw)$$

$\hat{y} = Hw$

$$y - Hw = (y - \hat{y}) = \begin{bmatrix} \text{residual}_1 \\ \text{residual}_2 \\ \vdots \\ \text{residual}_N \end{bmatrix}$$

$\text{residual}_i = y_i - \hat{y}_i$

$$\begin{array}{|c|c|c|c|} \hline \text{residual}_1 & \text{residual}_2 & \cdots & \text{residual}_N \\ \hline \end{array} \quad \begin{array}{|c|} \hline \text{residual}_1 \\ \text{residual}_2 \\ \vdots \\ \text{residual}_N \\ \hline \end{array}$$

$$= \begin{bmatrix} \text{residual}_1^2 + \\ \text{residual}_2^2 + \\ \cdots \\ \text{residual}_N^2 \end{bmatrix} = \sum_{i=1}^{N} \text{residual}_i^2 = RSS(w)$$

Computing the gradient of RSS

Step 3: Take the gradient

$$\nabla RSS(w) = \nabla[(y-Hw)^T(y-Hw)] \quad \{\text{Gradient of RSS}$$

$$= -2H^T(y-Hw)$$

Why? By analogy to 1D case:

$$\frac{\partial}{\partial w}(y-hw)(y-hw) = \frac{\partial}{\partial w}(y-hw)^2 = 2\cdot(y-hw)^1(-h)$$

$$= -2h(y-hw)$$

(scalar)

Approach 1: closed form solution

Step 4: Approach 1: Set the gradient $= 0$

Closed-form solution

$$\nabla RSS(w) = -2H^T(y-Hw) = 0$$

Solve for w:

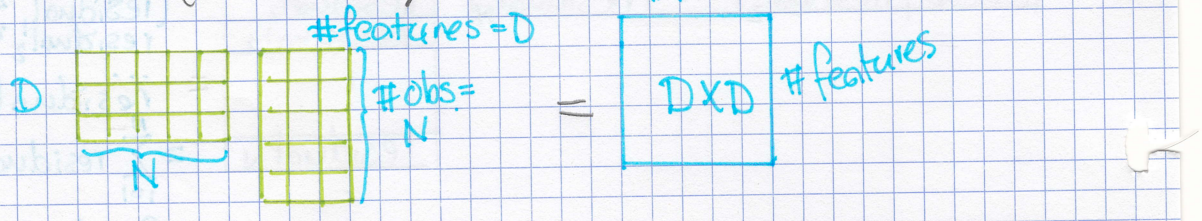
$$-2H^Ty + 2H^TH\hat{w} = 0$$

$$H^TH\hat{w} = H^Ty$$

$$\underbrace{(H^TH)^{-1}H^TH}_{I}\hat{w} = (H^TH)^{-1}H^Ty$$

$$\hat{w} = (H^TH)^{-1}H^Ty$$

$$\hat{w} = (H^TH)^{-1}H^Ty$$



Invertible if:
In most cases is $N > D$
(really # of linearly ind. observations)

Complexity of inverse:
$O(D^3)$

Approach 2: Gradient descent
Step 4 approach 2

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \underbrace{\nabla RSS(w^{(t)})}_{-2H^T(y - Hw)}$$

$$w^{(t)} + 2\eta H^T \underbrace{(y - Hw^{(t)})}_{\hat{y}(w^{(t)})}$$

Feature-by-Feature update

$$RSS(w) = \sum_{i=1}^{N} (y_i - h(x_i)^T w)^2$$

$$= \sum_{i=1}^{N} (y_i - w_0 h_0(x_i) - w_1 h_1(x_i) - \ldots - w_D h_D(x_i))^2$$

Partial with respect to w_j

$$\sum_{i=1}^{N} 2(y_i - w_0 h_0(x_i) - w_1 h_1(x_i) - \ldots - w_D h_D(x_i))^1 \cdot (-h_j(x_i))$$

$$= -2 \sum_{i=1}^{N} h_j(x_i)(y_i - h(x_i)^T w)$$

update to j^{th} feature weight:

$$w_j^{(t+1)} \leftarrow w_j^{(t)} - \eta\left(-2\sum_{i=1}^{N} h_j(x_i)\left(y_i - \underbrace{h^T(x_i)\,w^{(t)}}_{\hat{y}_i}\right)\right)$$

j^{th} feature: # bathrooms

$$w_j^{(t+1)} \leftarrow w_j^{(t)} + 2\eta\sum_{i=1}^{N} h_j(x_i)\left(y_i - \hat{y}_i(w^{(t)})\right)$$

(# bath for house i)

if underestimating impact of # bath ($\hat{w}_j^{(t)}$ is too small) then $\left(y_i - \hat{y}_i(w^{(t)})\right)$ on average weighted by # bath will be positive $\Rightarrow w_j^{(t+1)} > w_j^{(t)}$ (increase)

Summary of gradient descent for multiple regression

init $w^{(1)} = 0$ (or randomly, or smartly), $t = 1$

while $\|\nabla RSS(w^{(t)})\| > \varepsilon$ (ε: tolerance)

  for $j = 0, \ldots, D$

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

    $w_j^{(t+1)} \leftarrow w_j^{(t)} - \eta \, partial[j]$

  $t \leftarrow t + 1$

$\sqrt{partial[0]^2 + \ldots + partial[D]^2}$