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*

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

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

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

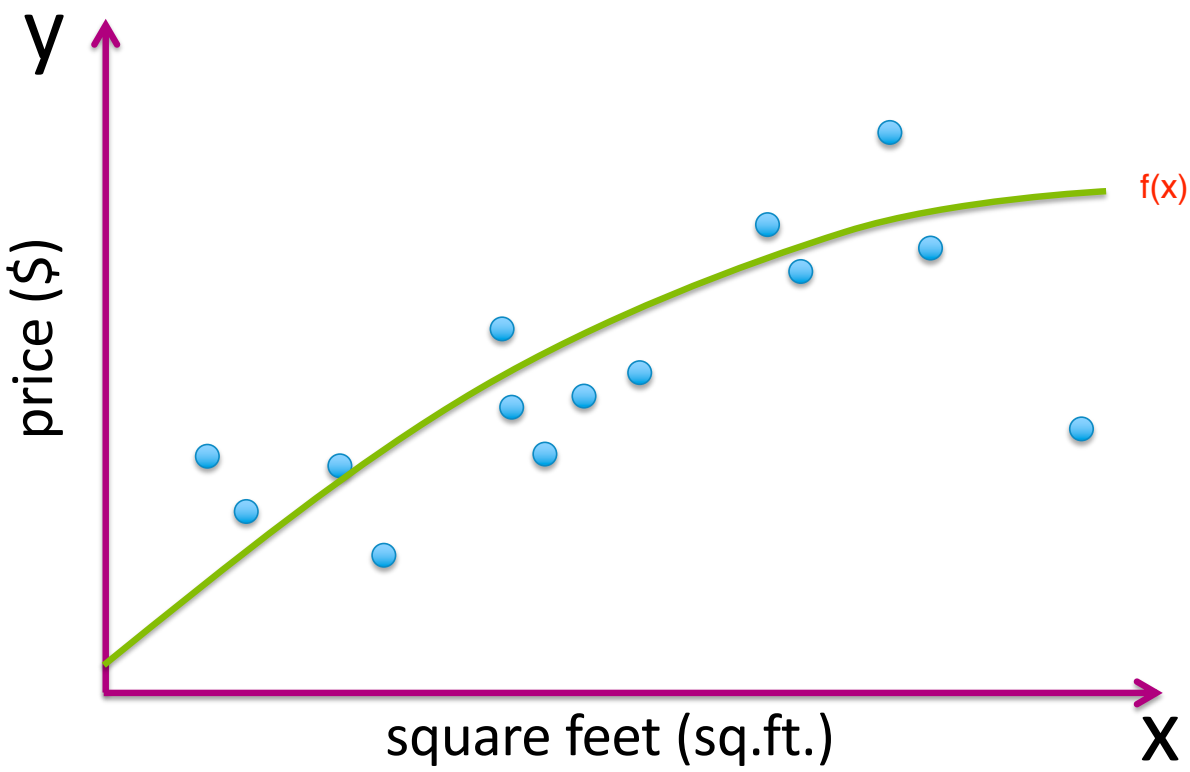
$(x_4 = \text{sq.ft.}, y_4 = \$)$

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

Input vs. Output:

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

Model –
How we *assume* the world works



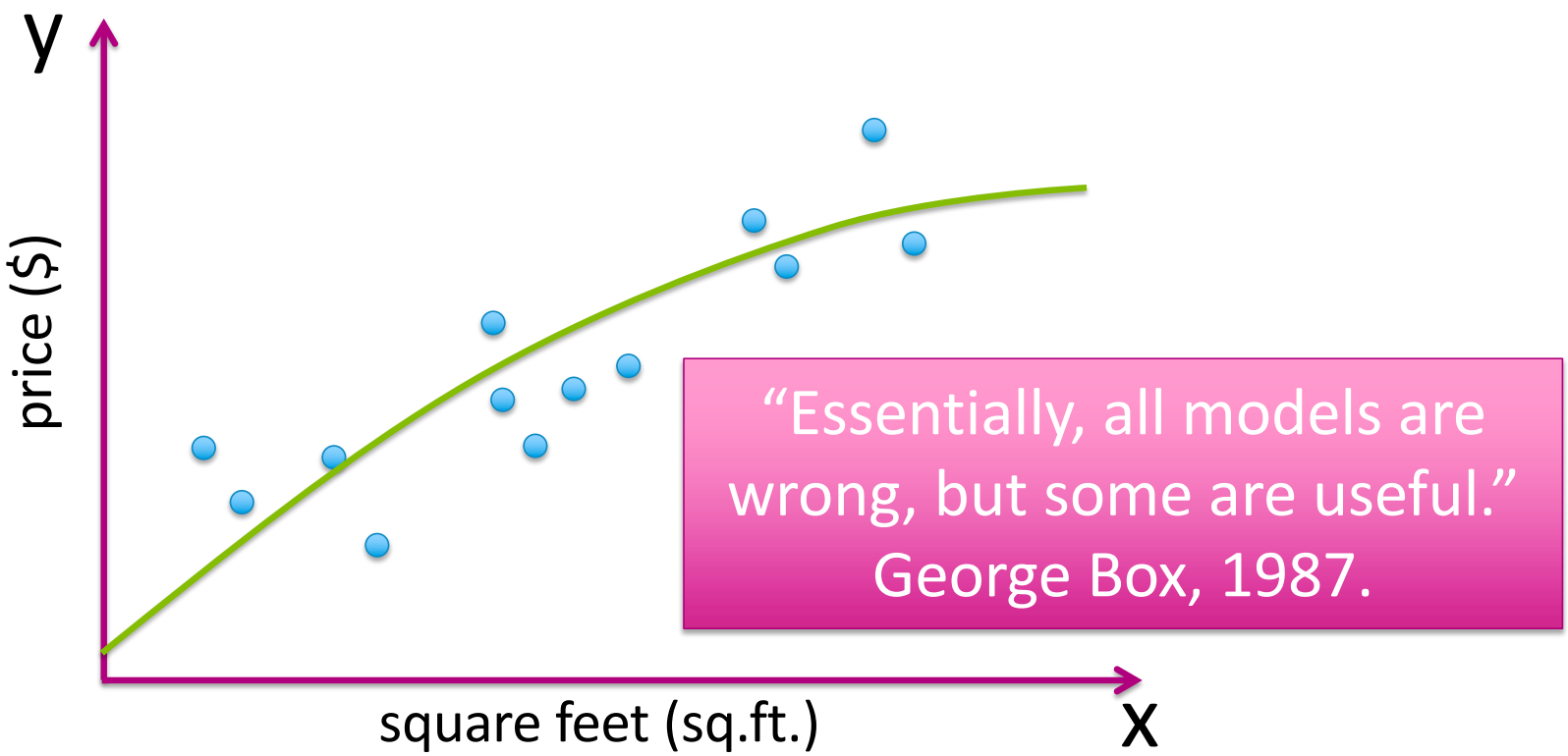
f(x) = expected relationship between x and y

Regression model:

$y_i = f(x_i) + \text{Error}_i$

$E[\text{Error}_i] = 0$

Model –
How we *assume* the world works



"Essentially, all models are wrong, but some are useful."
George Box, 1987.

Process

- Chose a model

  a family of functions that we feel best captures the dynamics of the system.

  { f_w(w) | (w1, w2, …., wn) in Rn} —> a function parameterized by parameters w1, w2, …, wn in R

- Find the best fit to our data set.

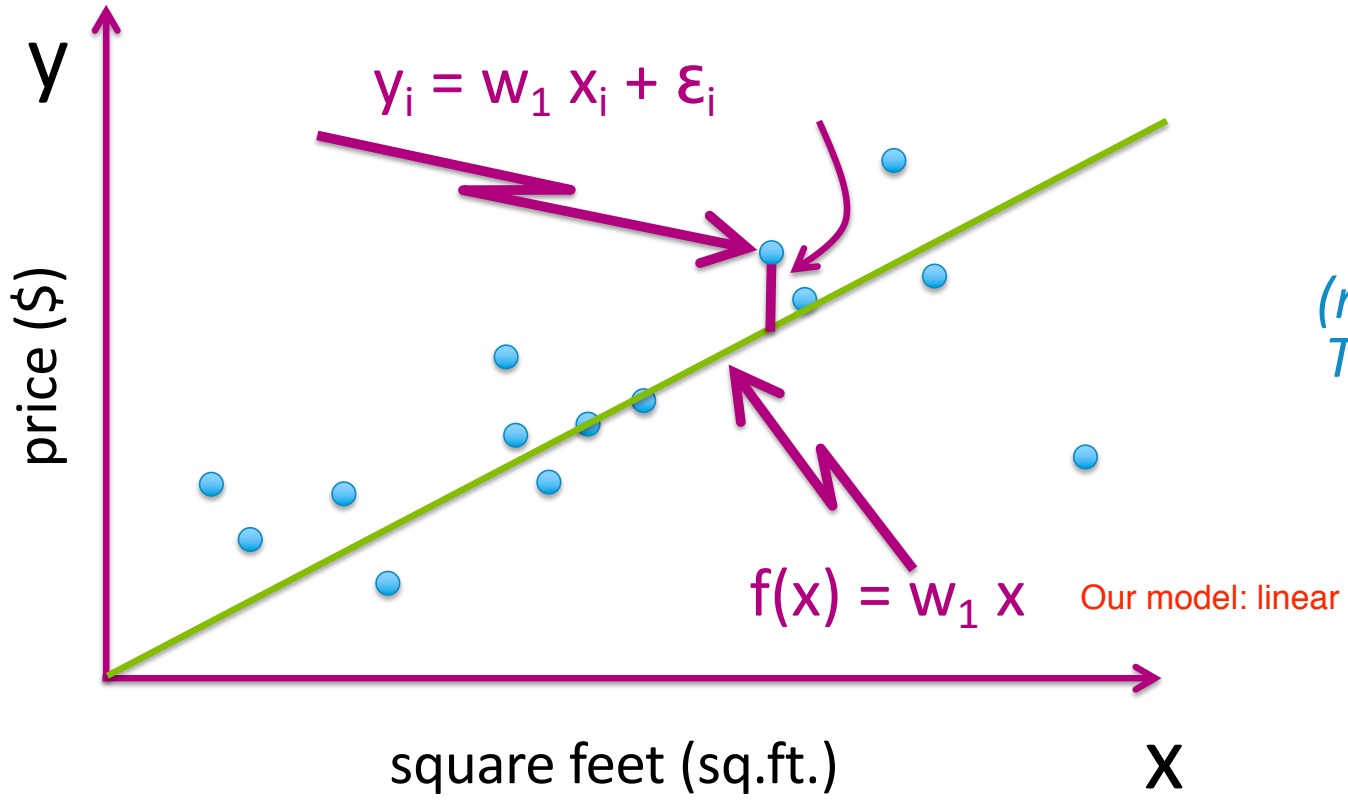
  Which function in family is best? What are the best choice of parameters?

  Find w* that meets some sort of condition we find ideal in family of function; i.e. minimizes or maximizes some quantity.

- Use the fitted function to make future predictions.

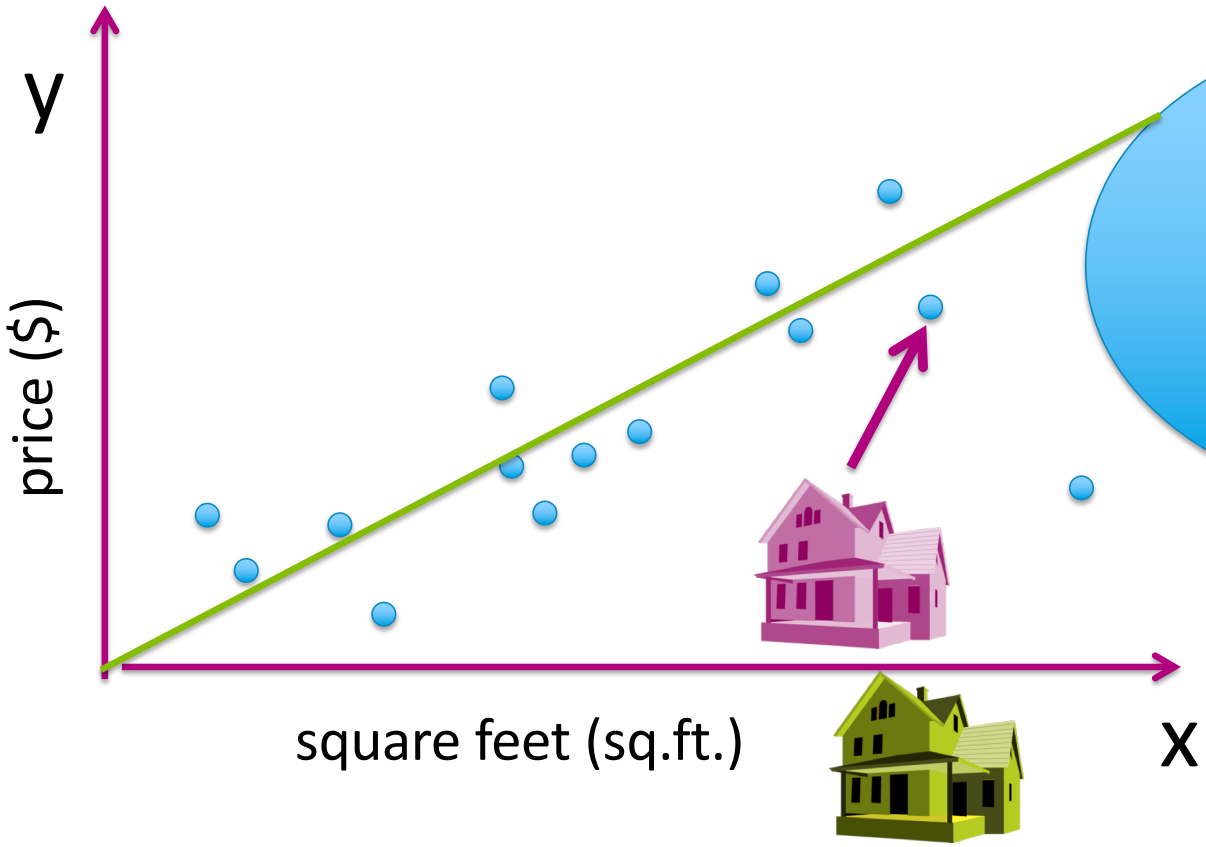
  y_future = f(x_future) + Error

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

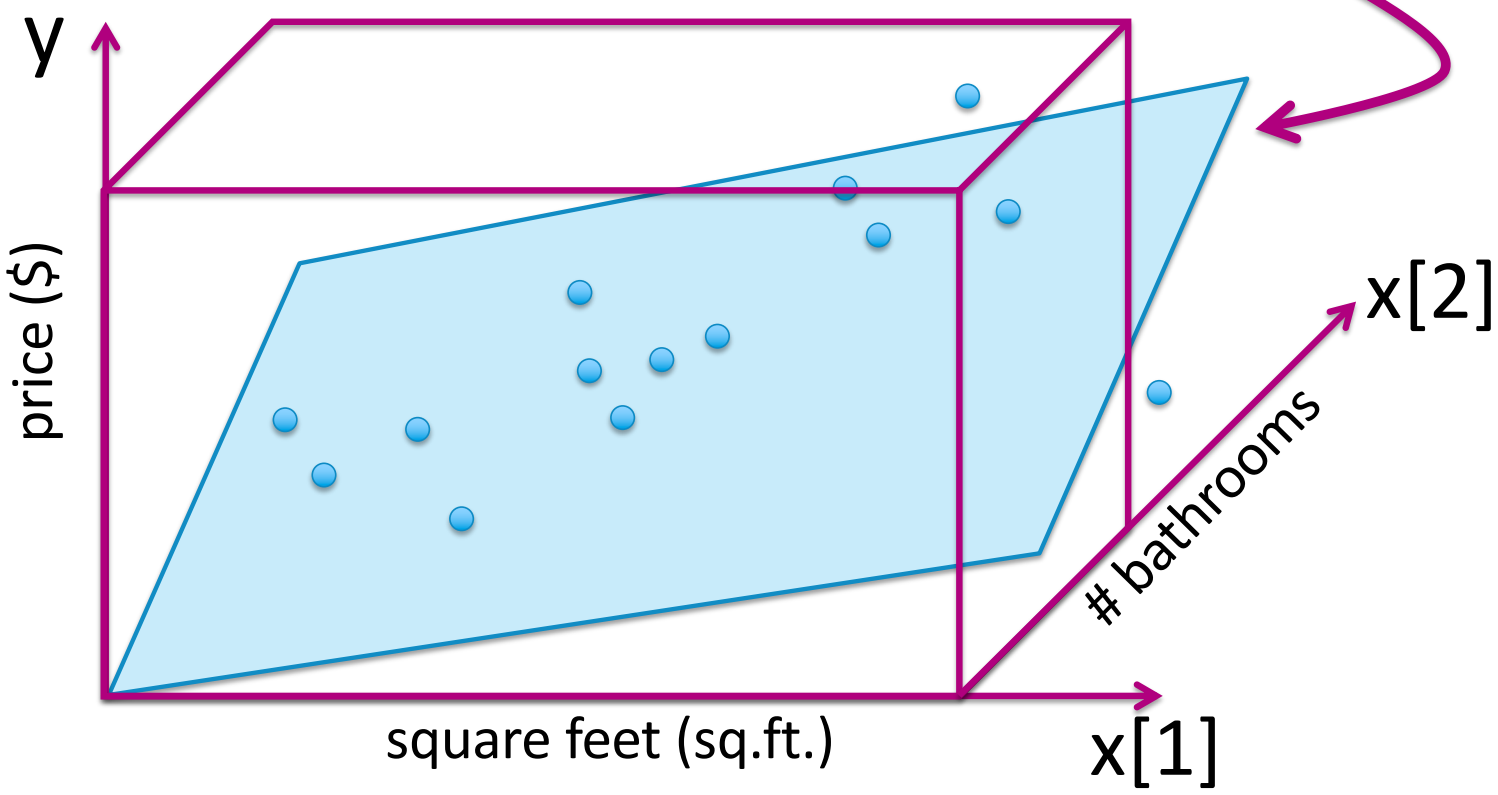


Only 1 bathroom! Obviously price not the same as it would be for house with 3 bathrooms

square footage may not be a suitable predictor alone

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
- …

General notation

Output: y ← scalar

Inputs: **x** = (x[1],x[2],…, x[d])

↖ d-dim vector

e.g., x[1] = sq. ft, x[2] = #baths and so on.

Notational conventions:

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

$\mathbf{x}_i$ = 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 $\mathbf{x}_i = (x_i[1], x_i[2], \ldots, x_i[d])$

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

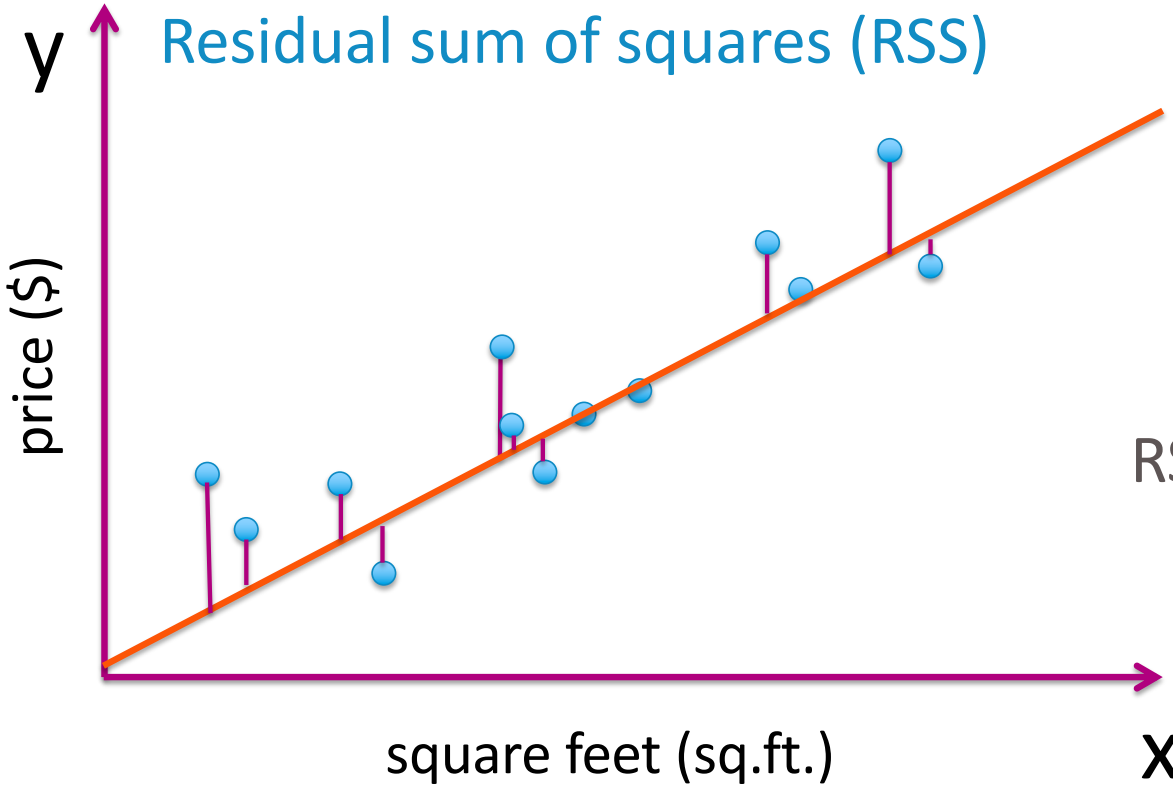
$= \sum_{j=1}^{d} w_j x_i[j] + \varepsilon_i \qquad = \mathbf{w}^T \mathbf{x}_i + \varepsilon_i = \mathbf{x}_i^T \mathbf{w} + \varepsilon_i$

feature 1 = x[1] = sq. ft.
feature 2 = x[2] = #bath

…
feature d = x[d] =lot size

Fitting the linear regression model

“Cost” of using a given line

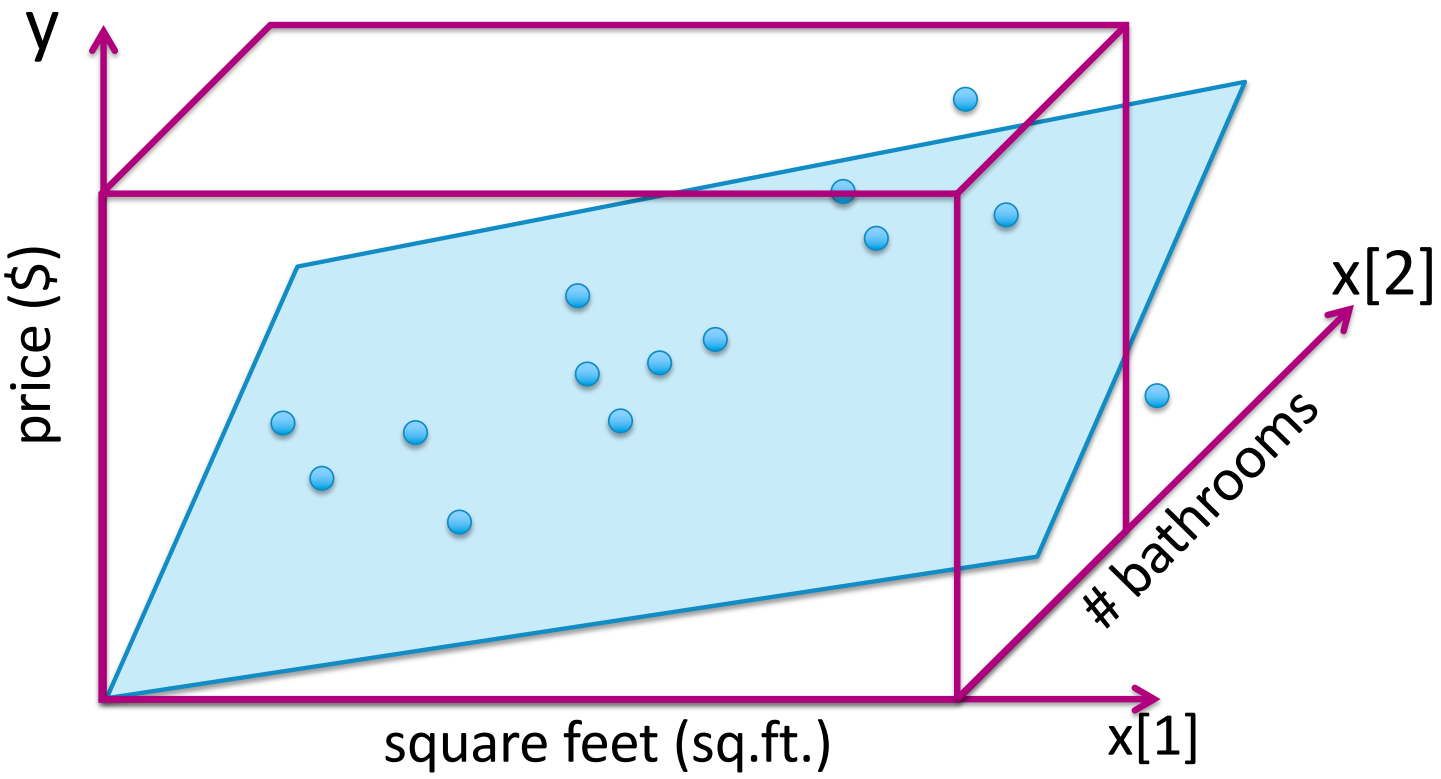


Sum squared error

$$\text{RSS}(w_1) = \sum_{i=1}^{N} (y_i - w_1 x_i)^2$$

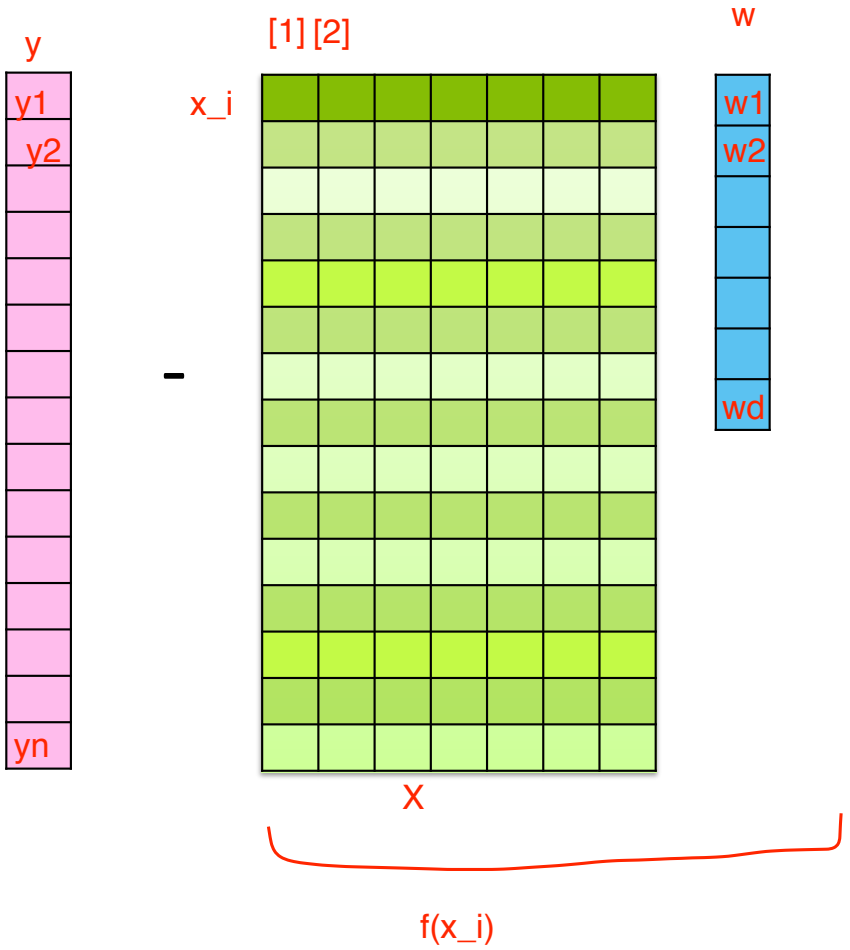
RSS for multiple regression

RSS(w)



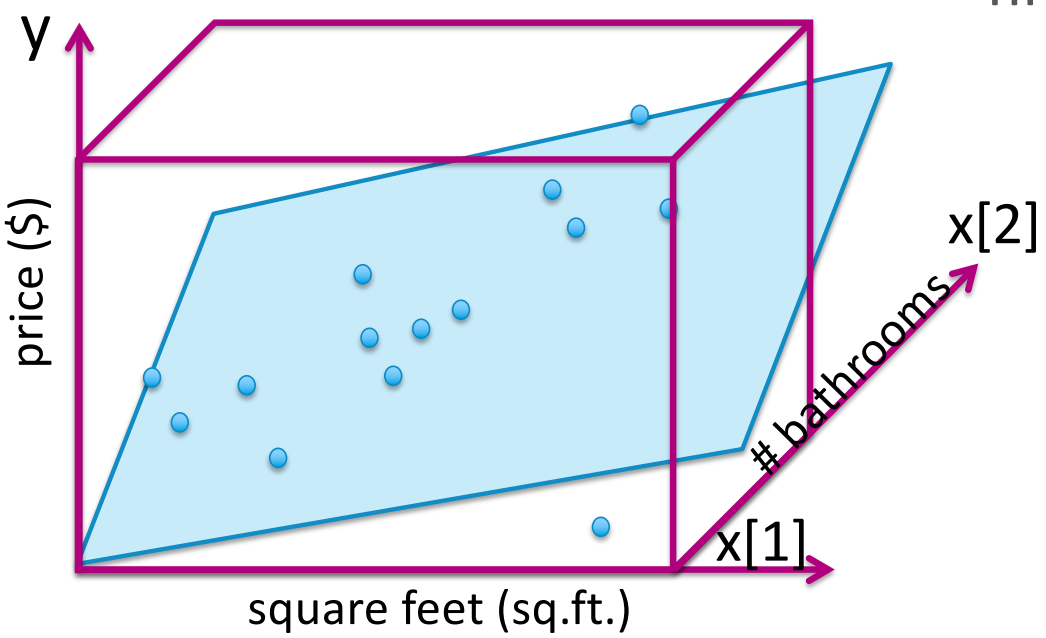
Rewrite in matrix notation

For all observations together



RSS(w) = sum_i=1^n (yi - xi^Tw = (y-Xw))

RSS for multiple regression



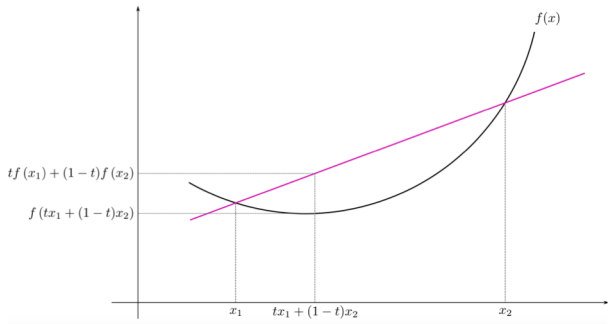
Objective: Find the best fit, i.e., find the **w** that minimizes

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

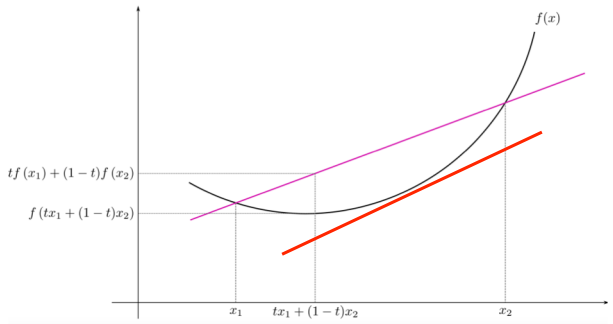
Interlude: Optimization (and convex functions)

Minimizing a univariate function

Convex functions $f''(x) > 0$ for all w,



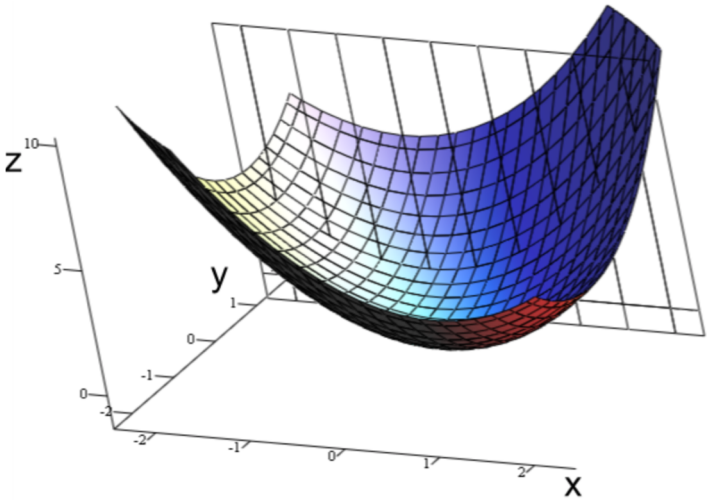
line between any two points always overestimates f(x) between those two points



at any point the tangent plane will always underestimate the function values at other points

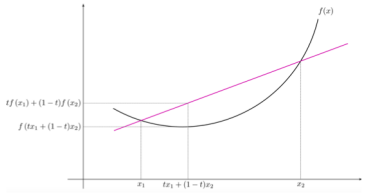
Minimizing a multivariate function

Convex functions



gradient f = 0 —> global minimum

Any two points on surface the line between them overestimates height of surface.

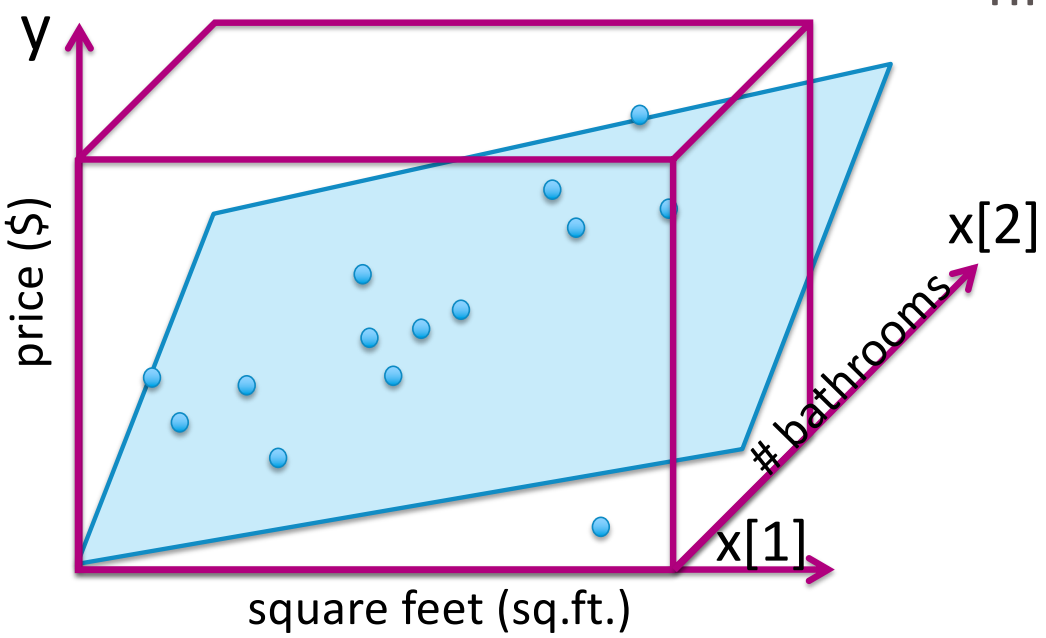


Tangent plane always underestimates surface and only touches at tangent point

f(w0) + (w-w0)^T gradient f(w0) <= f(w) for all w
tangent plane at w0

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

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

RSS (**w**) is a convex function of **w**

Take the second partial derivative with respect to w_i and note that it is always > 0

1. Solve for $\nabla RSS(w) = 0$

Gradient of RSS

$$\nabla RSS(w) = \nabla[(y-Xw)^T(y-Xw)]$$

Closed form solution

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

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

Solution: ``normal equations'' $X^TX\hat{w} = X^Ty$

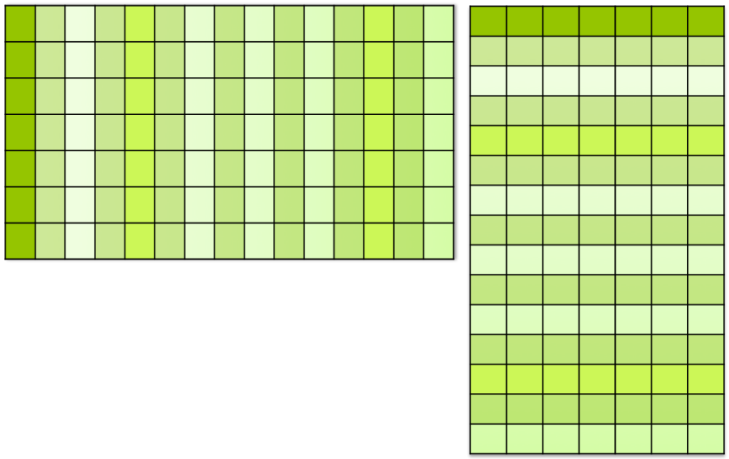
could solve this with an inverse, but
it might exist, doing so is slow, and
there are better ways of going about doing this.
We know there is a solution

Geometric intuition for solution

Again, back to getting closed form solution.

Solution to normal equations $X^T X \hat{w} = X^T y$

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



Invertible if:

Complexity of inverse:

2. Gradient descent

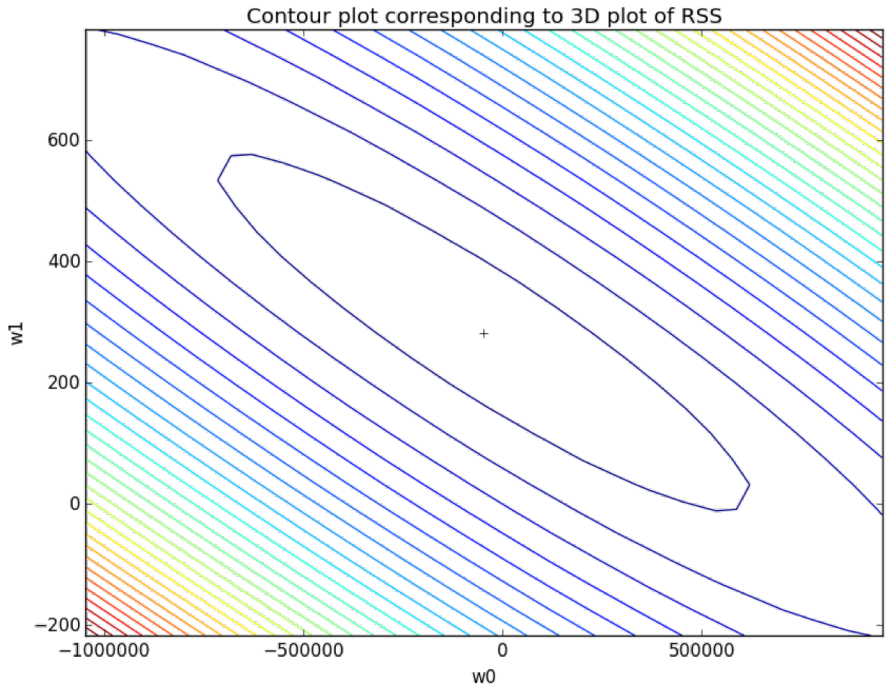
Gradient Descent – univariate case

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

Gradient Descent – multivariate case

Gradient descent for linear regression:

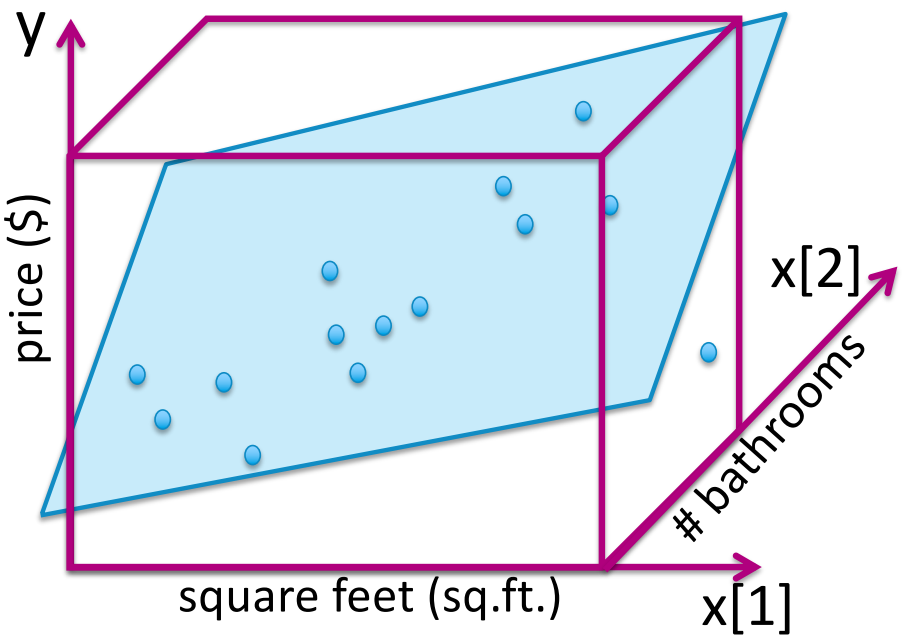
repeatedly move in direction of negative gradient



while not converged

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

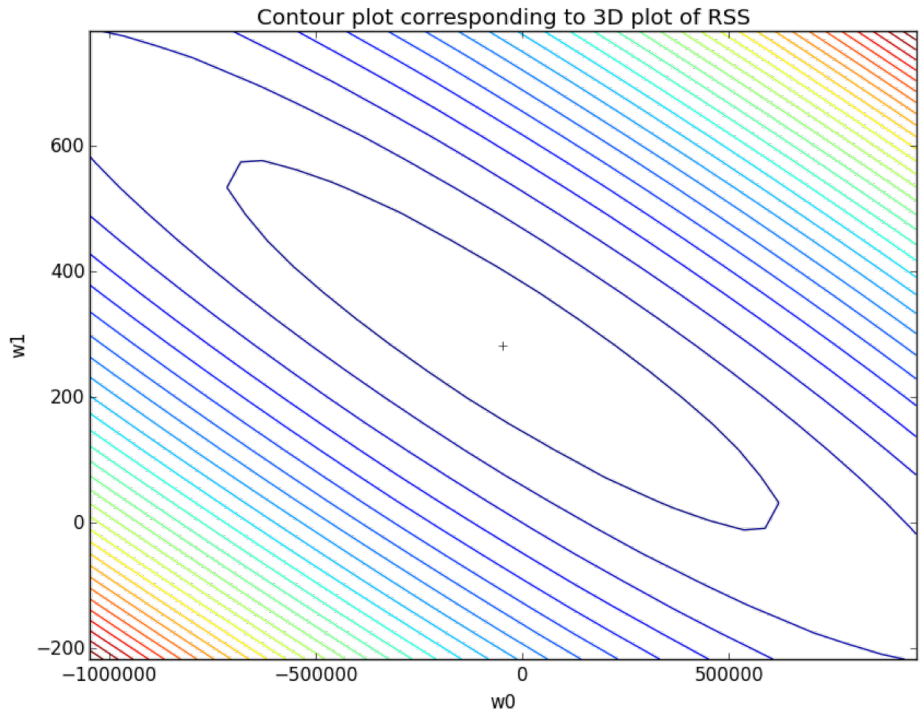
Interpreting elementwise



Update to j^{th} feature weight:

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

Summary of gradient descent for multiple regression



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

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

  for j=1,…,d

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

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

  t ← 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 **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 $\frac{1}{n}\sum_i \boldsymbol{x}_i = \mathbf{0}$ (*) then solution is simple.

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

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

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

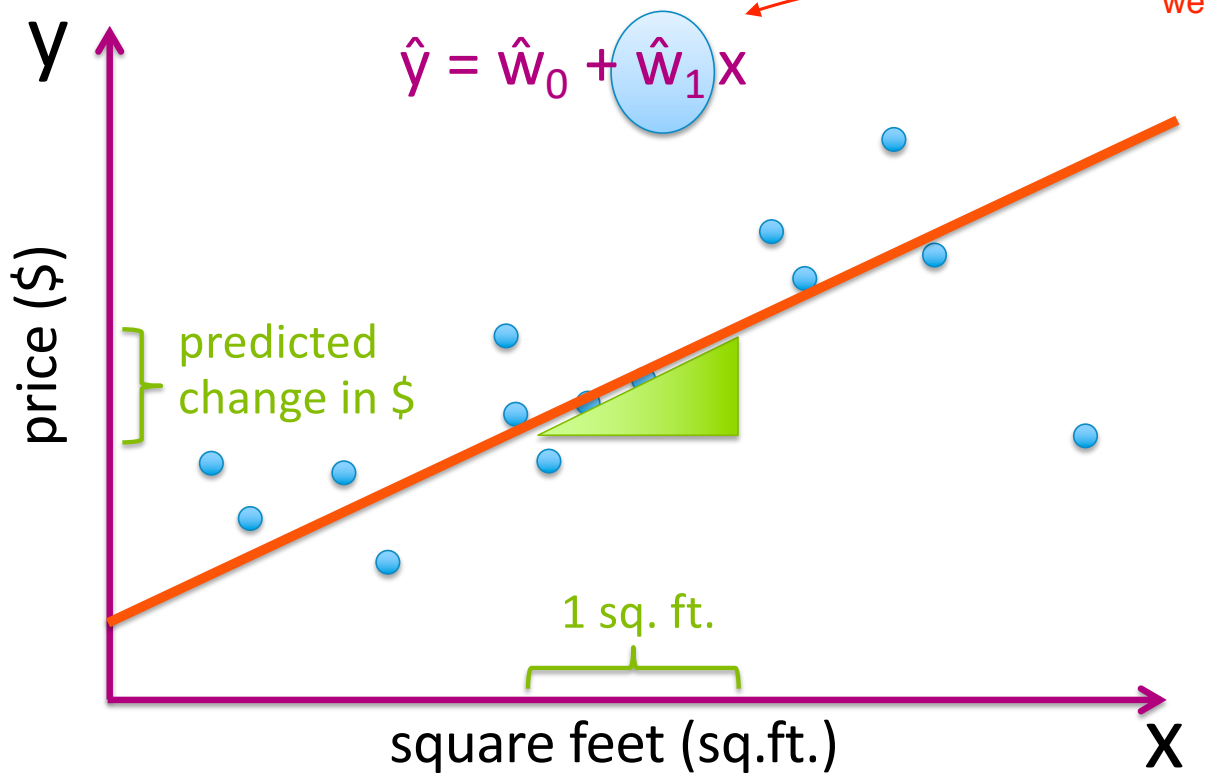
2. Show how to transform, aka ``demean" any linear regression problem so that (*) holds.

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

Interpreting the fitted function

Interpreting the coefficients – Simple linear regression

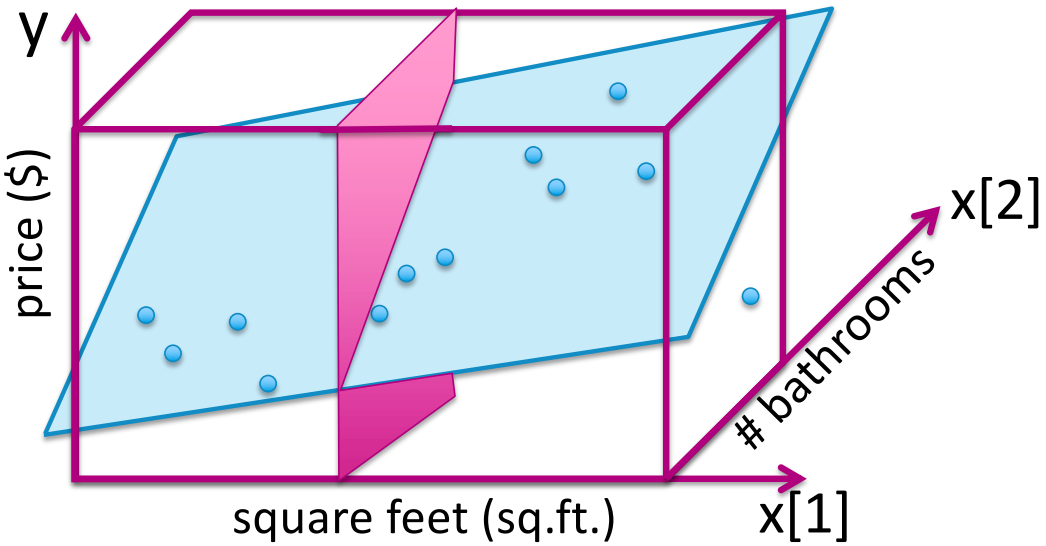
How much do we expect the price to increase as we increase the square feet by one unit.



Interpreting the coefficients – Two linear features

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

fix

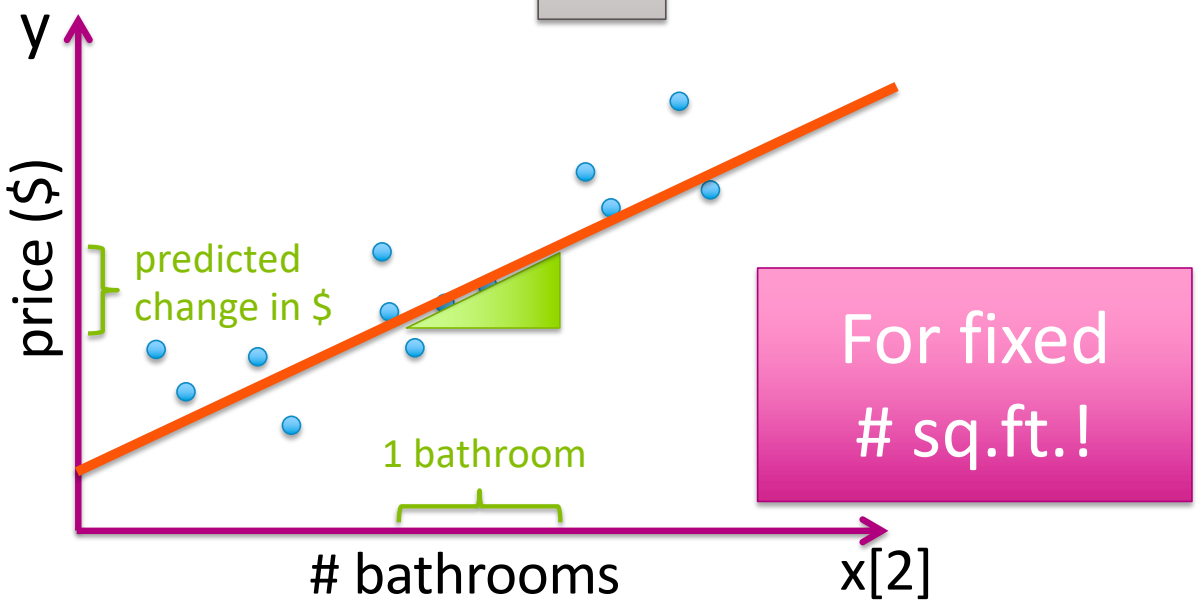


w2 = Fixing x[1] , i..e the square footage, how much does increasing x[2], number of bathrooms, by one unit increase the price.

Interpreting the coefficients – Two linear features

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

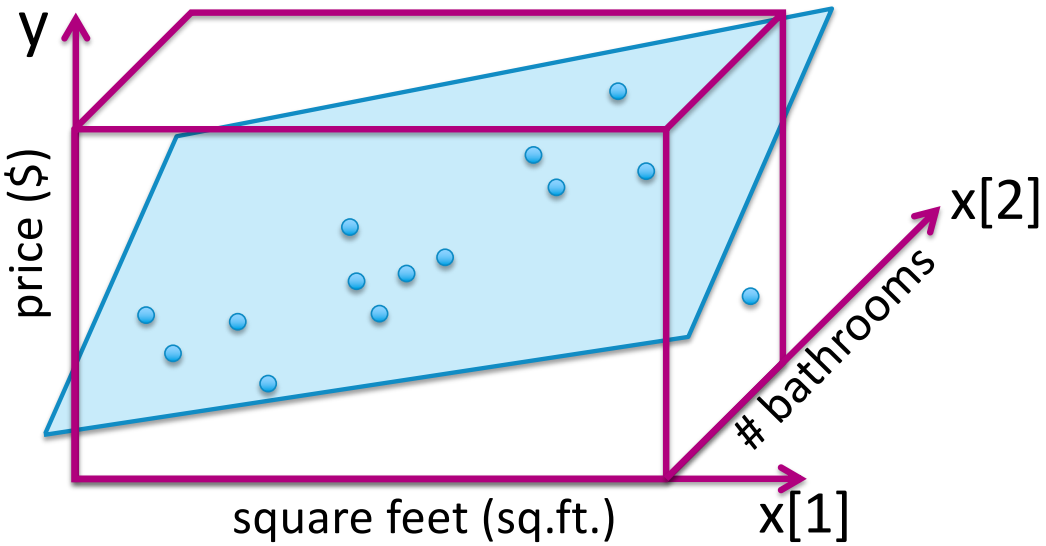
fix



Interpreting the coefficients – Multiple linear features

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

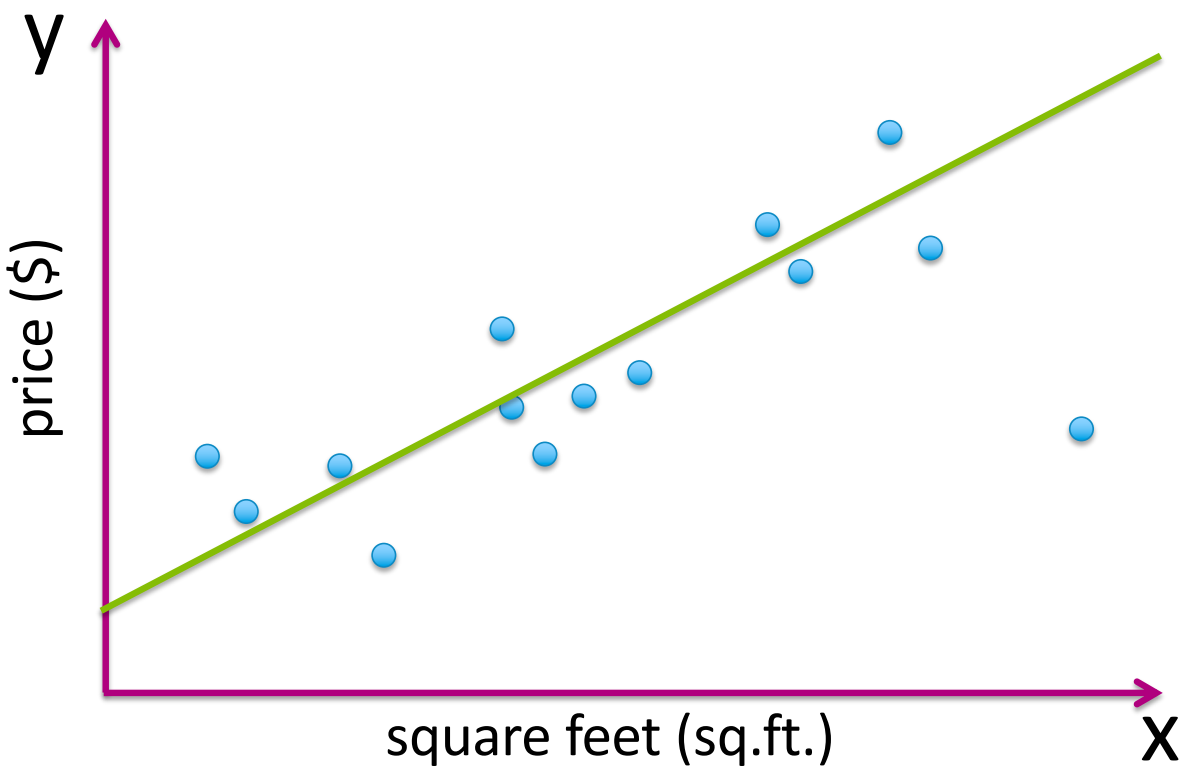
fix fix



Why min RSS?

Makes sense for gaussian noise…

Assuming Gaussian noise



Model for ε_i:

N(0, sigma^2)

Implied distribution on y_i:

yi ~ N(wTxi , sigma^2)

Maximum likelihood estimate of params

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

Maximum Likelihood estimate

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

P(y | x, w, sigma^2) = product_{i=1}^n phi(yi ; xTw, sigma^2)
where phi is the probability density function of the normal distribution

find the maximum of this function; note is not strictly a probability so this equality is not quite true
but phi is still a relative measure of probability so it performs adequately.

taking the log we see that the likelihood

Conclusion

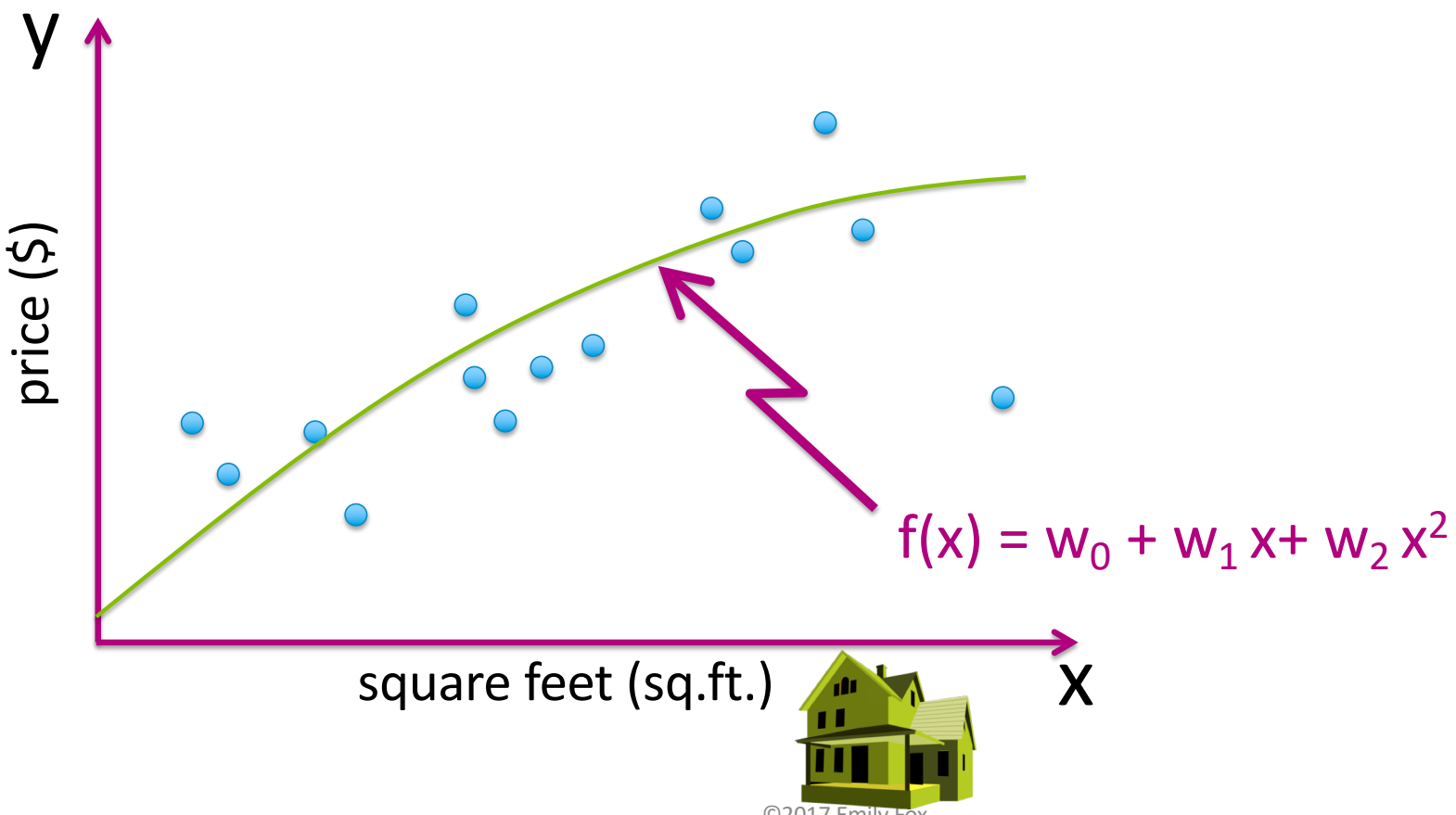
If $y_i = x_i^T \mathbf{w} + \varepsilon_i$ is $N(x^T w, \sigma^2)$

Then choosing w to maximize log-likelihood

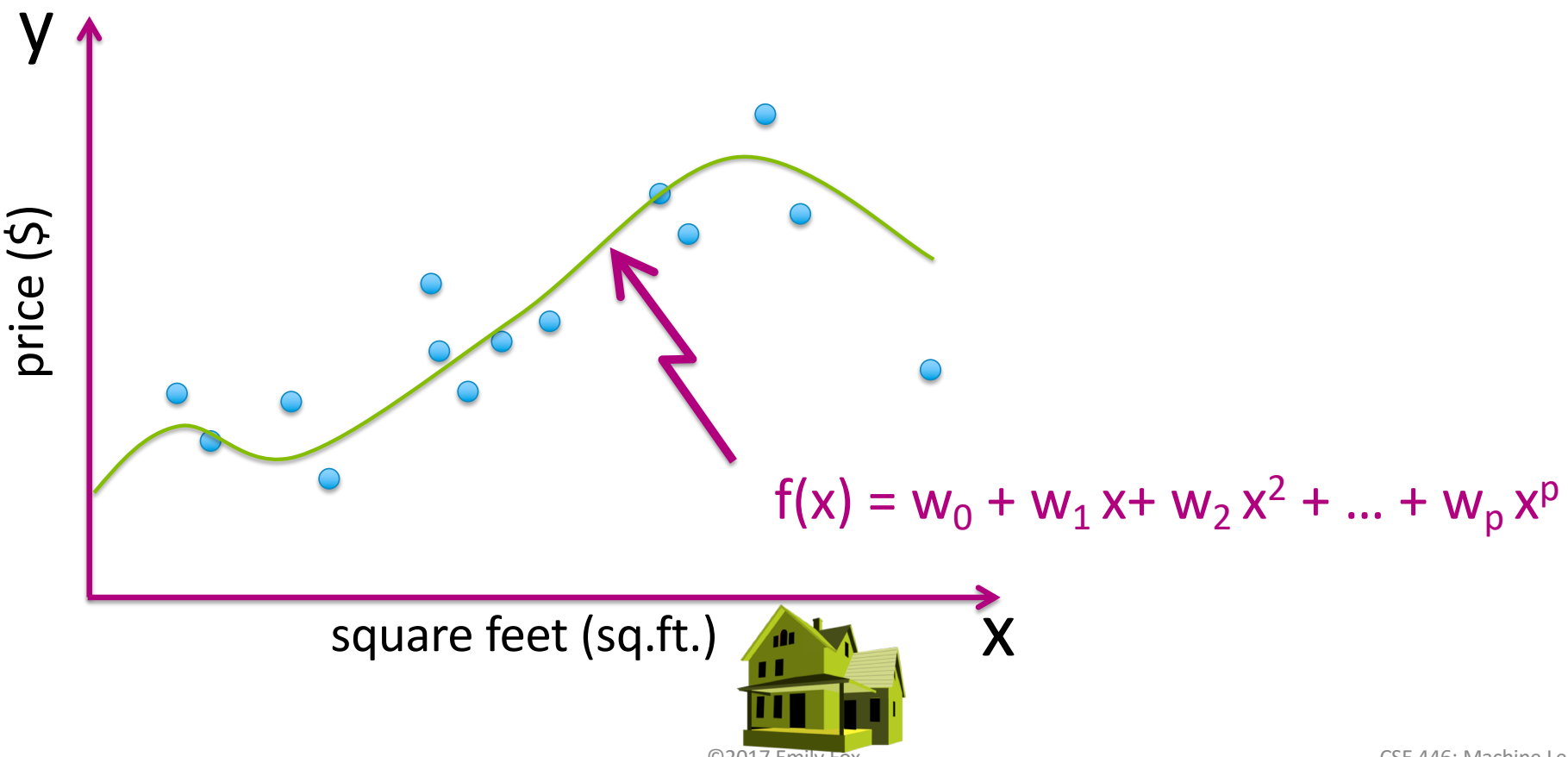
is same as choosing w to minimize RSS!

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 $h(x_i)$.

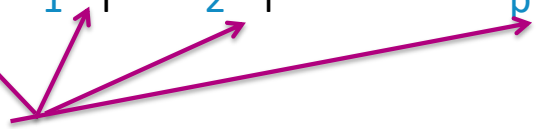
Example: x_i

$h_0(x_i)$	$h_1(x_i)$	$h_2(x_i)$	$h_3(x_i)$	$h_4(x_i)$	$h_5(x_i)$
1	x_i	x_i^2	x_i^3	x_i^4	x_i^5

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 transformed inputs 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

Why might we want to use polynomial regression?

- Taylor Series!

More generally

- 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 $h(\mathbf{x}_i)$.

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

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

General notation

Output: y ← scalar

Inputs: **x** = (x[1],x[2],…, x[d]) ← d-dim vector

Notational conventions:

- $\mathbf{x}_i$ = input of i^{th} data point (*vector*)
- $x_i[j]$ = j^{th} input of i^{th} data point (*scalar*)
- $h(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_d(\mathbf{x}))$ feature map applied to input x (*vector*)
- $h_j(\mathbf{x})$ = 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],…, 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 $\hat{w}$ that minimizes 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