COMS 4721: Machine Learning for Data Science
Lecture 2, 1/19/2017

Prof. John Paisley

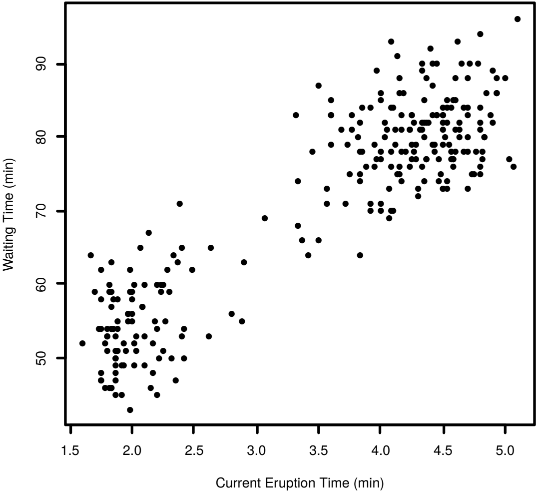
Department of Electrical Engineering
& Data Science Institute

Columbia University

LINEAR REGRESSION

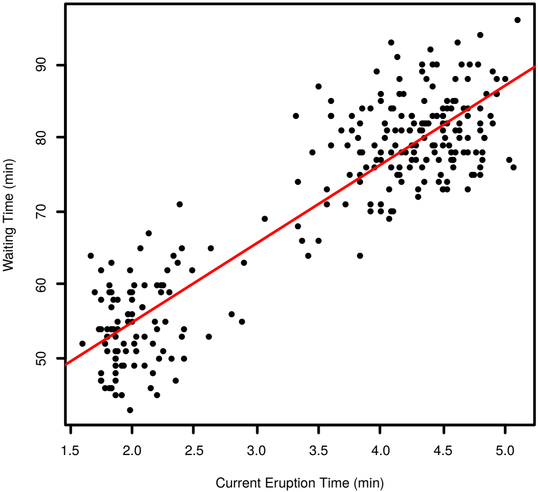
Example: Old Faithful

Example: Old Faithful



Can we meaningfully predict the time between eruptions only using the duration of the last eruption?

Example: Old Faithful



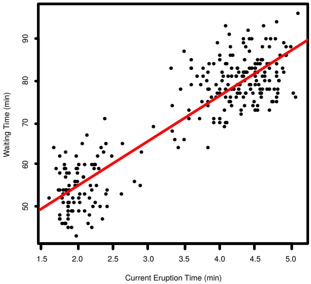
Can we meaningfully predict the time between eruptions only using the duration of the last eruption?

Example: Old Faithful

One model for this

(wait time) $\approx w_0 +$ (last duration) $\times w_1$
(simply learning a line to fit the data.)

- w_0 and w_1 are to be learned.
- This is an example of linear regression.



Refresher

w_1 is the slope, w_0 is called the intercept, bias, shift, offset.

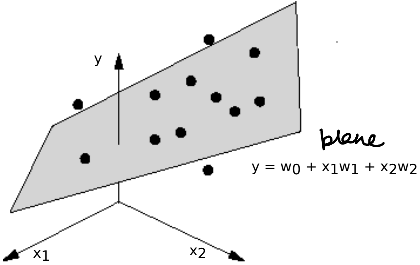
Higher dimensions

Inputs are 2-dimensional.
Output is 1-dimensional.

Two inputs

(output) $\approx w_0 + (\text{input } 1) \times w_1 + (\text{input } 2) \times w_2$

With two inputs the intuition is the same $\longrightarrow$



Regression: Problem Definition

Data

Input: $x \in \mathbb{R}^d$ (i.e., measurements, covariates, features, indepen. variables)
Output: $y \in \mathbb{R}$ (i.e., response, dependent variable) ↳ which is a real valued nos

Goal

maps x to y (i/p to o/p)

Find a function $f : \mathbb{R}^d \to \mathbb{R}$ such that $y \approx f(x; w)$ for the data pair (x, y).
$f(x; w)$ is called a *regression function*. Its free parameters are w.

Definition of linear regression

A regression method is called *linear* if the prediction f is a linear function of the unknown parameters w. (doesn't mean it is a linear function of x)

Least squares linear regression model

Model

The linear regression model we focus on now has the form

dot product
w_j with j^{th} dimension of x_j

$$y_i \approx f(x_i; w) = w_0 + \sum_{j=1}^{d} x_{ij} w_j.$$

basis (shifts the plane/line through the intercept.)

Model learning

We have the set of *training data* $(x_1, y_1) \ldots (x_n, y_n)$. We want to use this data to learn a w such that $y_i \approx f(x_i; w)$. But we first need an *objective function* to tell us what a "good" value of w is.

Least squares

The *least squares* objective tells us to pick the w that minimizes the sum of squared errors

training dataset

$$w_{\text{LS}} = \arg\min_{w} \sum_{i=1}^{n} (y_i - f(x_i; w))^2 \equiv \arg\min_{w} \mathcal{L}.$$

prediction

Least squares in pictures

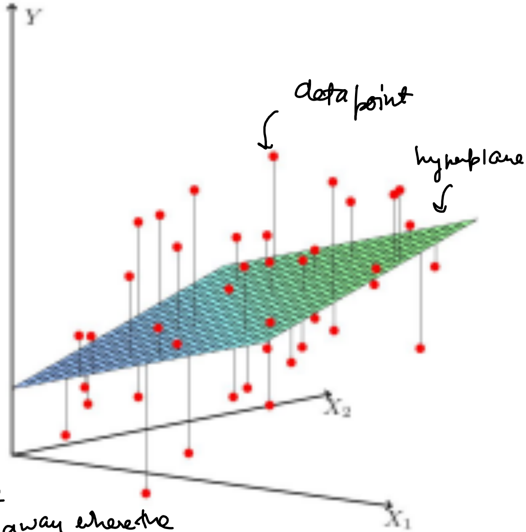
Observations:

Vertical length is error.

The objective function $\mathcal{L}$ is the sum of all the squared lengths.

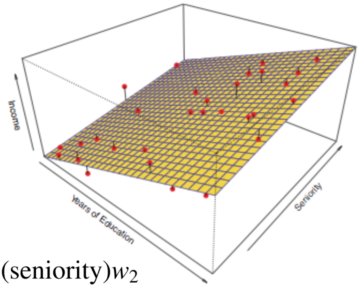
Find weights (w_1, w_2) plus an offset w_0 to minimize $\mathcal{L}$.

(w_0, w_1, w_2) defines this plane.



find an orientation for the plane so that it cuts through the data in a way where the sum of squares of these lengths are minimized.

Example: Education, Seniority and Income



2-dimensional problem

Input: (education, seniority) $\in \mathbb{R}^2$.

Output: (income) $\in \mathbb{R}$

Model: (income) $\approx w_0 + \text{(education)}w_1 + \text{(seniority)}w_2$

Question: Both $w_1, w_2 > 0$. What does this tell us?

Answer: As education and/or seniority goes up, income tends to go up.

(Caveat: This is a statement about correlation, not causation.)

Thus far

We have data pairs (x_i, y_i) of measurements $x_i \in \mathbb{R}^d$ and a response $y_i \in \mathbb{R}$. We believe there is a linear relationship between x_i and y_i,

$$y_i = w_0 + \sum_{j=1}^{d} x_{ij} w_j + \epsilon_i$$

error [approximation of y_i]

and we want to minimize the objective function

$$\mathcal{L} = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - w_0 - \textstyle\sum_{j=1}^{d} x_{ij} w_j)^2$$

Goal: minimize the sum of squared errors.

with respect to $(w_0, w_1, \ldots, w_d)$.

Can math notation make this easier to look at/work with?

Notation: Vectors and Matrices

We think of data with d dimensions as a *column* vector:

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix} \quad \text{(e.g.)} \Rightarrow \quad \begin{bmatrix} \text{age} \\ \text{height} \\ \vdots \\ \text{income} \end{bmatrix}$$

A set of n vectors can be stacked into a matrix:

$$\mathbf{X} = \begin{bmatrix} x_{11} & \dots & x_{1d} \\ x_{21} & \dots & x_{2d} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{nd} \end{bmatrix} = \begin{bmatrix} - \, x_1^T \, - \\ - \, x_2^T \, - \\ \vdots \\ - \, x_n^T \, - \end{bmatrix}$$

Assumptions for now:

1 ▶ All features are treated as continuous-valued ($x \in \mathbb{R}^d$)

2 ▶ We have more observations than dimensions ($d < n$)

NOTATION: REGRESSION (AND CLASSIFICATION)

Usually, for linear regression (and classification) we include an intercept term w_0 that doesn't interact with any element in the vector $x \in \mathbb{R}^d$.

extend x from $\mathbb{R}^d$ to $\mathbb{R}^{d+1}$

It will be convenient to attach a 1 to the first dimension of each vector x_i (which we indicate by $x_i \in \mathbb{R}^{d+1}$) and in the first column of the matrix X:

$$x_i = \begin{bmatrix} 1 \\ x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1d} \\ 1 & x_{21} & \dots & x_{2d} \\ \vdots & & \vdots & \\ 1 & x_{n1} & \dots & x_{nd} \end{bmatrix} = \begin{bmatrix} 1 - & x_1^T & - \\ 1 - & x_2^T & - \\ & \vdots & \\ 1 - & x_n^T & - \end{bmatrix}.$$

We also now view $w = [w_0, w_1, \dots, w_d]^T$ as $w \in \mathbb{R}^{d+1}$.

LEAST SQUARES IN VECTOR FORM

Original least squares objective function: $\mathcal{L} = \sum_{i=1}^{n}(y_i - w_0 - \sum_{j=1}^{d} x_{ij}w_j)^2$

Using vectors, this can now be written: $\mathcal{L} = \sum_{i=1}^{n}(y_i - x_i^T w)^2$

why is not

$2x_i^T w y_i$

Least squares solution (vector version)

We can find w by setting,

$$\nabla_w \mathcal{L} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \nabla_w (y_i^2 - \underline{2w^T x_i y_i} + w^T x_i x_i^T w) = 0.$$

Solving gives,

$$-\sum_{i=1}^{n} 2y_i x_i + \Big(\sum_{i=1}^{n} 2x_i x_i^T\Big) w = 0 \quad \Rightarrow \quad w_{\text{LS}} = \Big(\sum_{i=1}^{n} x_i x_i^T\Big)^{-1}\Big(\sum_{i=1}^{n} y_i x_i\Big).$$

Gradient always column vector?

$X \to n \times d$ matrix

The i^{th} dimension of y is approximated by the i^{th} row of X times the $\vec{w}$.

Least squares solution (matrix version)

Least squares in matrix form is even cleaner.

Start by organizing the y_i in a column vector, $y = [y_1, \dots, y_n]^T$. Then

$$\mathcal{L} = \sum_{i=1}^{n}(y_i - x_i^T w)^2 = \|y - Xw\|^2 = (y - Xw)^T(y - Xw).$$

vector of errors of our approximation.

If we take the gradient with respect to w, we find that

$$\nabla_w \mathcal{L} = 2X^T Xw - 2X^T y = 0 \quad \Rightarrow \quad w_{\text{LS}} = (X^T X)^{-1} X^T y.$$

RECALL FROM LINEAR ALGEBRA

Recall: Matrix × vector $\left(X^T y = \sum_{i=1}^n y_i x_i\right)$

Taking each column of X & scaling it with corresponding value in vector y. & adding those up.

$$\begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = y_1 \begin{bmatrix} | \\ x_1 \\ | \end{bmatrix} + y_2 \begin{bmatrix} | \\ x_2 \\ | \end{bmatrix} + \cdots + y_n \begin{bmatrix} | \\ x_n \\ | \end{bmatrix}$$

Recall: Matrix × matrix $\left(X^T X = \sum_{i=1}^n x_i x_i^T\right)$

1st col. of left matrix X 1st row of right matrix and then adding all those up.

outer product.

$$\begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_n \\ | & | & & | \end{bmatrix} \begin{bmatrix} - x_1^T - \\ - x_2^T - \\ \vdots \\ - x_n^T - \end{bmatrix} = x_1 x_1^T + \cdots + x_n x_n^T.$$

LEAST SQUARES LINEAR REGRESSION: KEY EQUATIONS

Two notations for the *key equation*

vector w that minimizes the sum of squared errors

1

$$w_{\text{LS}} = \left(\sum_{i=1}^{n} x_i x_i^T\right)^{-1}\left(\sum_{i=1}^{n} y_i x_i\right) \quad \Longleftrightarrow \quad w_{\text{LS}} = (X^T X)^{-1} X^T y.$$

2

Making Predictions

We use w_{LS} to make predictions.

Given x_{new}, the least squares prediction for y_{new} is

$$y_{\text{new}} \approx x_{\text{new}}^T w_{\text{LS}}$$

(dot product of our i/p with our least square vector.)

LEAST SQUARES SOLUTION

assuming we can invert this matrix.

Potential issues

Calculating $w_{\text{LS}} = (X^T X)^{-1} X^T y$ assumes $(X^T X)^{-1}$ exists.

When doesn't it exist?

Answer: When $X^T X$ is not a full rank matrix.

we need $X^T X$ to be a full rank matrix

When is $X^T X$ full rank?

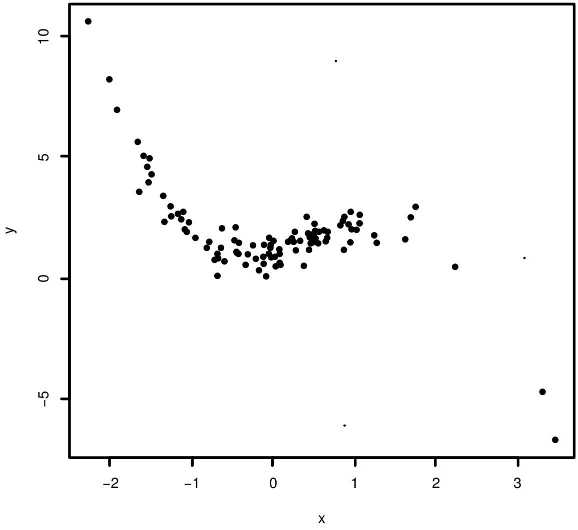
1 Answer: When the $n \times (d+1)$ matrix X has at least $d+1$ *linearly independent* rows. This means that any point in $\mathbb{R}^{d+1}$ can be reached by a weighted combination of $d+1$ rows of X.

(it's not possible to find d+1 linearly independent rows of x because there aren't that many rows of x in this case.

2 Obviously if $n < d+1$, we can't do least squares. If $(X^T X)^{-1}$ doesn't exist, there are an infinite number of possible solutions.

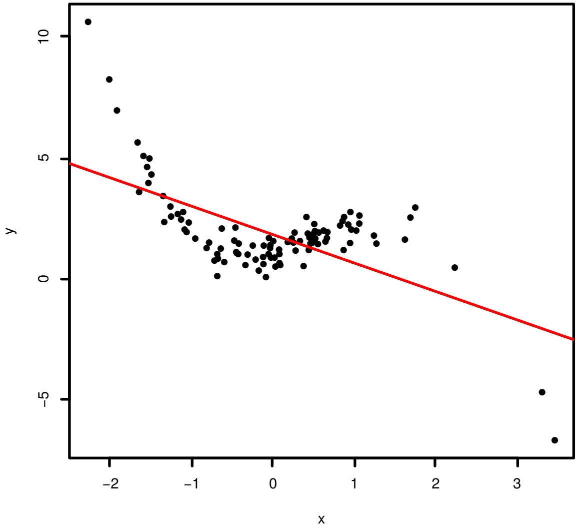
Takeaway: We want $n \gg d$ (i.e., X is "tall and skinny").

No. of data points is to be much greater than the dimensionality of the problem.


10
5
0
−5
y
−2
−1
0
1
2
3
x

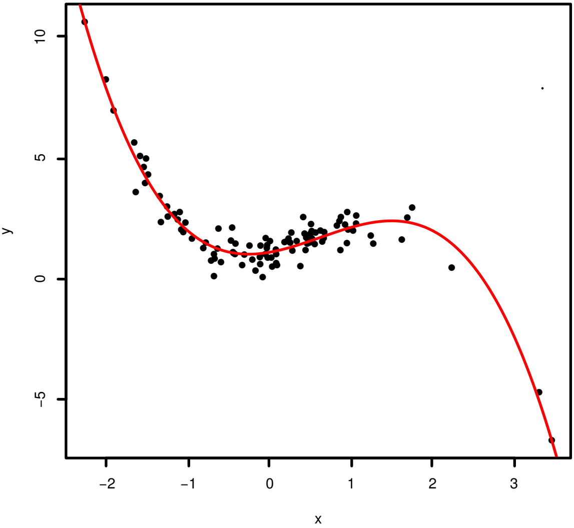
BROADENING LINEAR REGRESSION

$$y = w_0 + w_1 x$$



$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$$

3rd polynomial
we are taking
our data x &
performing a fn
on it to take it
from 1 dimensions
to 3 "



Recall: Definition of linear regression

A regression method is called *linear* if the prediction f is a linear function of the unknown parameters w.

[need not be linear in x.]

unknown parameter

- Therefore, a function such as $y = w_0 + w_1 x + w_2 x^2$ is *linear* in w. The LS solution is the same, only the preprocessing is different.
- E.g., Let $(x_1, y_1) \dots (x_n, y_n)$ be the data, $x \in \mathbb{R}$, $y \in \mathbb{R}$. For a pth-order polynomial approximation, construct the matrix

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & \dots & x_2^p \\ \vdots & & & \vdots & \\ 1 & x_n & x_n^2 & \dots & x_n^p \end{bmatrix}$$

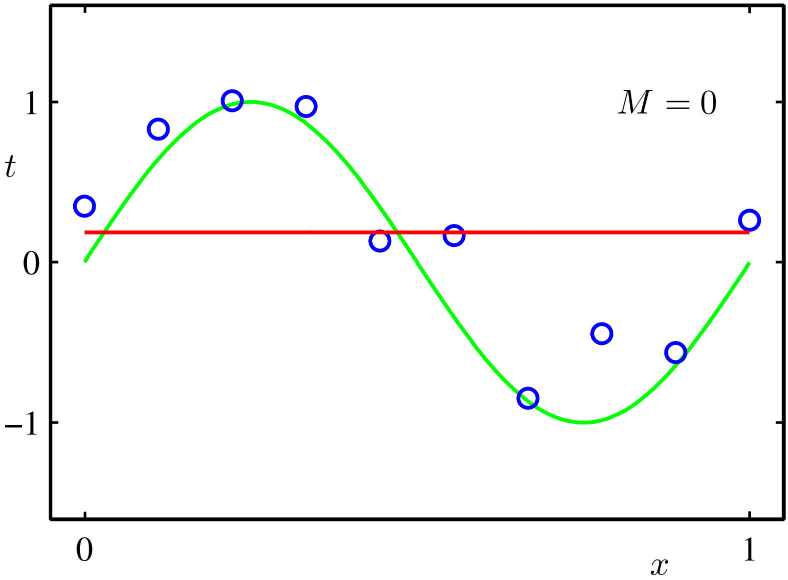
$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$

$y \approx Xw$

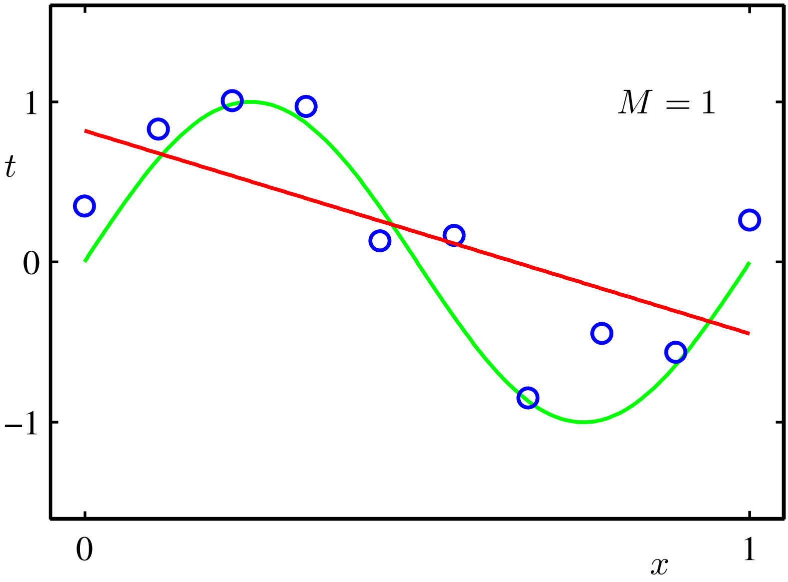
- Then solve exactly as before: $w_{\text{LS}} = (X^T X)^{-1} X^T y$.

↳ p+1 dimensional

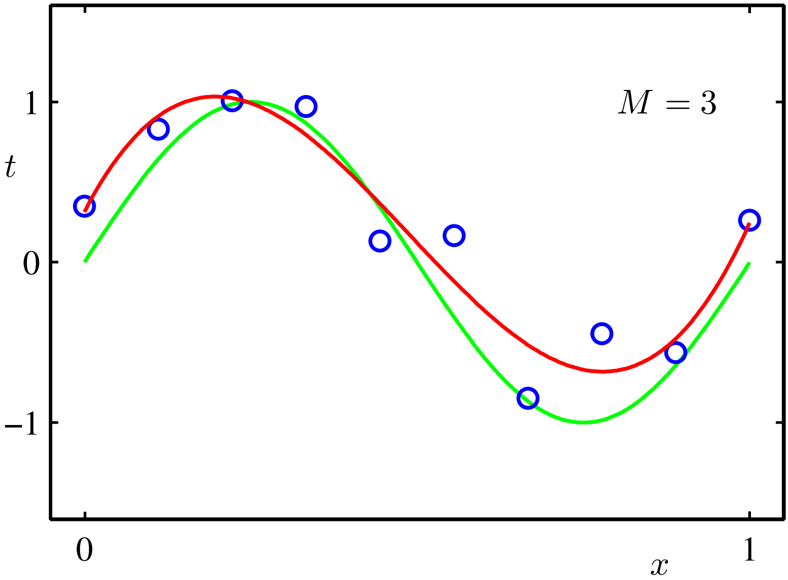
Polynomial regression (Mth order)



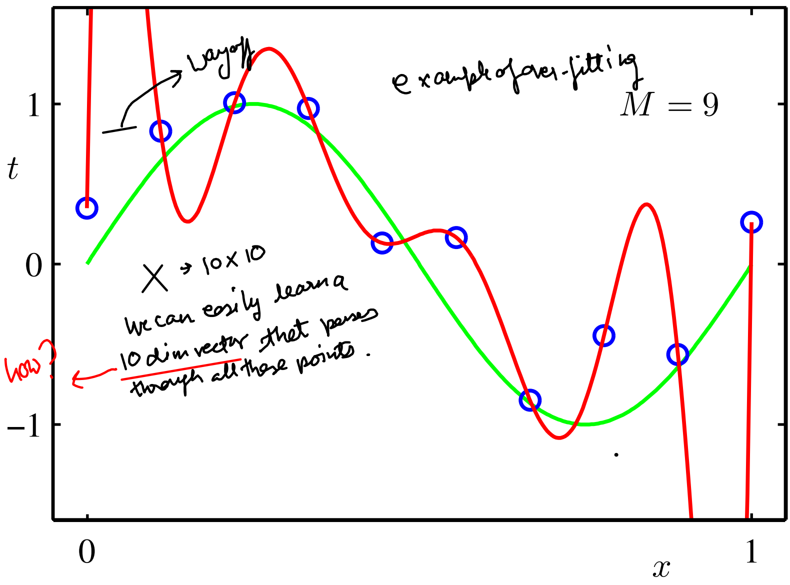
Polynomial regression (Mth order)



POLYNOMIAL REGRESSION (MTH ORDER)



POLYNOMIAL REGRESSION (MTH ORDER)



Example: 2nd and 3rd order polynomial regression in $\mathbb{R}^2$

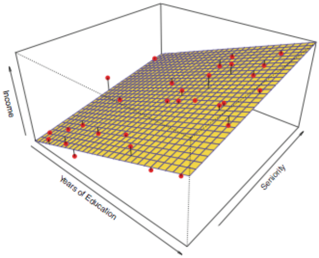
x of polynomial

The width of X grows as (order) $\times$ (dimensions) + 1. → for offset

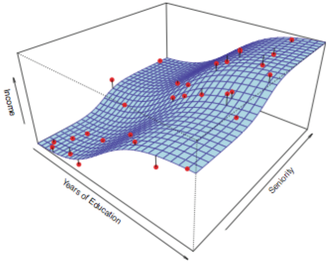
original problem

2nd order: $$y_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i1}^2 + w_4 x_{i2}^2$$

3rd order: $$y_i = w_0 + w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i1}^2 + w_4 x_{i2}^2 + \underline{w_5 x_{i1}^3} + \underline{w_6 x_{i2}^3}$$



(a) 1st order



(b) 3rd order

Further extensions

More generally, for $x_i \in \mathbb{R}^{d+1}$ least squares linear regression can be performed on functions $f(x_i; w)$ of the form

$$y_i \approx f(x_i, w) = \sum_{s=1}^{S} g_s(x_i) w_s.$$

For example,

$$\begin{aligned} g_s(x_i) &= x_{ij}^2 \\ g_s(x_i) &= \log x_{ij} \\ g_s(x_i) &= \mathbb{I}(x_{ij} < a) \\ g_s(x_i) &= \mathbb{I}(x_{ij} < x_{ij'}) \end{aligned}$$

indicator

As long as the function is *linear* in $w_1, \dots, w_S$, we can construct the matrix X by putting the transformed x_i on row i, and solve $w_{\text{LS}} = (X^T X)^{-1} X^T y$.

One caveat is that, as the number of functions increases, we need more data to avoid overfitting.

GEOMETRY OF LEAST SQUARES REGRESSION

* We take vector X_j, j^{th} column of X. Then scale it in some way according to W_j. So if it's W_j's +ve, we strech it or shrink it, but keep the direction same. If w_j is −ve, then we flip it's direction and stretch it or shrink it. We do this for each column of the matrix X and then we add these vectors up. We trying to make the sum of these vectors as close to Y as possible, which is a point in $\mathbb{R}^n$.

Thinking geometrically about least squares regression helps a lot.

- We want to minimize $\|y - Xw\|^2$. Think of the vector y as a point in $\mathbb{R}^n$. We want to find w in order to get the product Xw close to y.

- If X_j is the jth *column* of X, then $Xw = \sum_{j=1}^{d+1} w_j X_j$. *

- That is, we weight the columns in X by values in w to approximate y.

- The LS solutions returns w such that Xw is as close to y as possible in the Euclidean sense (i.e., intuitive "direct-line" distance).

GEOMETRY OF LEAST SQUARES REGRESSION

y is a vector $\in \mathbb{R}^3$, meaning we have 3 datapoints.

$$\arg\min_{w} \|y - Xw\|^2 \quad \Rightarrow \quad w_{\text{LS}} = (X^T X)^{-1} X^T y.$$

w_1, w_2 instead of x_1, x_2?

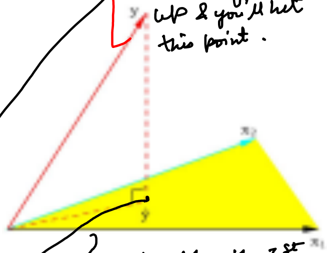
Take the Ist dimension multiply it by x_1, the second dimension by x_2. Sum of those up & you'll hit this point.

The columns of X define a $d+1$-dimensional subspace in the higher dimensional $\mathbb{R}^n$.

The closest point in that subspace is the *orthonormal projection* of y into the *column space* of X.

IInd column of the big matrix

Right: $y \in \mathbb{R}^3$ and data $x_i \in \mathbb{R}$.
$X_1 = [1, 1, 1]^T$ and $X_2 = [x_1, x_2, x_3]^T$



↳ vector of 1s, the Ist column of X.

The approximation is $\hat{y} = X w_{\text{LS}} = X(X^T X)^{-1} X^T y$.

X.

That's the closest that we can get the sum of those 2 vectors to o/p Y.

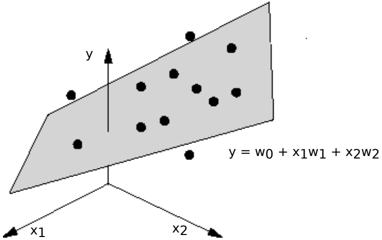
what is each axis here?

GEOMETRY OF LEAST SQUARES REGRESSION

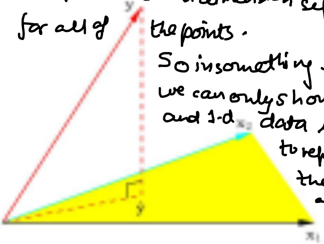
We have different values of Y, the o/p acc. to i/p

We stack all of the o/ps together & we represent each dimension separately for all of the points.

So in something like this, we can only show 3 points and 1-d data if we want to represent the DC offset.



(a) $y_i \approx w_0 + x_i^T w$ for $i = 1, \ldots, n$



(b) $y \approx Xw$

There are some key difference between (a) and (b) worth highlighting as you try to develop the corresponding intuitions.

(a) Can be shown for all n, but only for $x_i \in \mathbb{R}^2$ (not counting the added 1).

(b) This corresponds to $n = 3$ and one-dimensional data: $X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix}$.