**Foundations of Data Science, Fall 2020**

# 3. Linear Regression

**Prof. Dan Olteanu**

DaST Data • (Systems+Theory)

University of Zurich

Sept 22, 2020

https://lms.uzh.ch/url/RepositoryEntry/16830890400

https://uzh.zoom.us/j/96690150974?pwd=cnZmMTduWUtCeWoxYW85Z3RMYnpTZz09

---

**Russ Salakhutdinov**
9 October at 21:59

I am teaching a Deep Learning graduate course this Fall at CMU with over 300 MSc and PhD students enrolled.

Today, after our midterm, I received the following anonymous feedback: "Did I take the wrong exam? Does this exam cover too little machine learning stuff and focus too much on mathematics?"

I guess there is a common belief that Deep Learning is all about installing TensorFlow or PyTorch and training a gigantic convnet on multiple GPUs 😀

Like Comment Share

Csaba Szepesvari, Anima Anandkumar and 1.3k others



---

## Outline

### Goals

- Review the supervised learning setting
- Describe the linear regression framework
- Apply the linear model to make predictions
- Derive the least squares estimate

### Supervised Learning Setting

- Data consists of input and output pairs
- Inputs (also covariates, independent variables, predictors, features)
- Output (also variates, dependent variable, targets, labels)



---

## Why study linear regression?

- Least squares is at least 200 years old going back to Legendre and Gauss
- Francis Galton (1886): “Regression to the mean”
- Often real processes can be approximated by linear models
- More complex models require understanding linear regression
- Closed form analytic solutions can be obtained
- Many key notions of machine learning can be introduced



---

## Toy Example: Commute Times

Want to predict commute time into city centre

What variables would be useful?

- Distance to city centre
- Day of the week

### Data

| dist (km) | day | commute time (min) |
|---|---|---|
| 2.7 | fri | 25 |
| 4.1 | mon | 33 |
| 1.0 | sun | 15 |
| 5.2 | tue | 45 |
| 2.8 | sat | 22 |



---

## Linear Models

Suppose the input is a vector $\mathbf{x} \in \mathbb{R}^D$ and the output is $y \in \mathbb{R}$.

We have data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

Notation: data dimension $D$, size of dataset $N$, column vectors

**Linear Model**

$$\underbrace{y = w_0}_{\text{Bias/intercept}} + x_1 w_1 + \cdots + x_D w_D + \underbrace{\epsilon}_{\text{Noise/uncertainty}}$$

## Linear Models: Commute Time

**Linear Model**

$$y = w_0 + x_1 w_1 + \cdots + x_D w_D + \epsilon$$

Bias/intercept  Noise/uncertainty

Input encoding: mon-sun has to be converted to a number

- monday: 0, tuesday: 1, . . . , sunday: 6  BAD encoding!
- One-hot encoding: Use seven 0-1 features instead
- Simplifying example: 0 if weekend, 1 if weekday

Say $x_1 \in \mathbb{R}$ (distance) and $x_2 \in \{0, 1\}$ (weekend/weekday)

Linear model for commute time

$$y = w_0 + x_1 w_1 + x_2 w_2 + \epsilon$$



## Linear Model : Adding a feature for bias term

| dist $x_1$ | day $x_2$ | commute time $y$ |
|---|---|---|
| 2.7 | fri | 25 |
| 4.1 | mon | 33 |
| 1.0 | sun | 15 |
| 5.2 | tue | 45 |
| 2.8 | sat | 22 |

$\Longleftrightarrow$

| one $x_0$ | dist $x_1$ | day $x_2$ | commute time $y$ |
|---|---|---|---|
| 1 | 2.7 | fri | 25 |
| 1 | 4.1 | mon | 33 |
| 1 | 1.0 | sun | 15 |
| 1 | 5.2 | tue | 45 |
| 1 | 2.8 | sat | 22 |

**Model**

$$y = w_0 + x_1 w_1 + x_2 w_2 + \epsilon$$

**Model**

$$y = w_0 x_0 + x_1 w_1 + x_2 w_2 + \epsilon$$
$$= \mathbf{x} \cdot \mathbf{w} + \epsilon$$



## Learning Linear Models

Data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$

Model parameter $\mathbf{w}$, where $\mathbf{w} \in \mathbb{R}^D$

Training phase: (learning/estimation $\mathbf{w}$ from data)

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ data $\longrightarrow$ Learning Algorithm $\longrightarrow$ $\mathbf{w}$ (estimate)

Testing/Deployment phase: (predict $\widehat{y}_{\text{new}} = \mathbf{x}_{\text{new}} \cdot \mathbf{w}$)

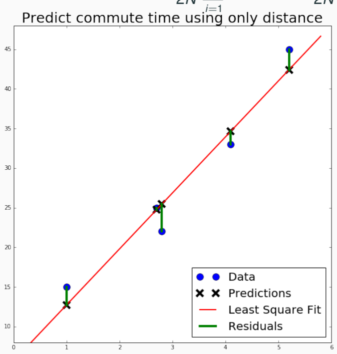
- How different is $\widehat{y}_{\text{new}}$ from $y_{\text{new}}$ (actual observation)?
- We should keep some data aside for testing before deploying a model



## Least Squares Objective Function

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $x_i, y_i \in \mathbb{R}$  $\widehat{y}(x) = w_0 + x \cdot w_1$ (no noise term)

$$\mathcal{L}(\mathbf{w}) = \mathcal{L}(w_0, w_1) = \frac{1}{2N} \sum_{i=1}^{N} (\widehat{y}_i - y_i)^2 = \frac{1}{2N} \sum_{i=1}^{N} (w_0 + x_i \cdot w_1 - y_i)^2$$




Loss function
Cost function
Objective Function
Energy Function
Notation - $\mathcal{L}, J, E, R$

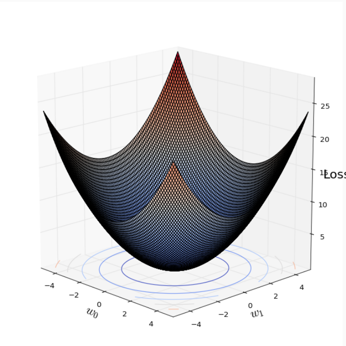
This objective is known as the residual sum of squares or (RSS)

The estimate $(w_0, w_1)$ is known as the least squares estimate



## Least Squares Objective Function

$$\mathcal{L}(\mathbf{w}) = \mathcal{L}(w_0, w_1) = \frac{1}{2N} \sum_{i=1}^{N} (w_0 + x_i \cdot w_1 - y_i)^2$$






$$\mathcal{L}(w_0, w_1) = \frac{1}{2N} \cdot \sum_i (w_0 + x_i \cdot w_1 - y_i)^2 \quad \arg\min_{w_0, w_1} \mathcal{L}(w_0, w_1)$$

Partial derivatives:

$$\frac{\partial \mathcal{L}}{\partial w_0} = \cancel{2} \cdot \frac{1}{\cancel{2}N} \underline{\sum_i} (w_0 + x_i \cdot w_1 - y_i) = 0$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = \cancel{2} \cdot \frac{1}{\cancel{2}N} \cdot \sum_i (w_0 + x_i \cdot w_1 - y_i) \cdot x_i = 0$$

Normal equations:

$$\begin{cases} w_0 \cdot \cancel{\frac{\sum_i 1}{N}} + w_1 \cdot \underbrace{\frac{\sum x_i}{N}}_{\bar{x}} - \underbrace{\frac{\sum y_i}{N}}_{\bar{y}} = 0 \\ w_0 \cdot \underbrace{\frac{\sum_i x_i}{N}}_{\bar{x}} + w_1 \cdot \frac{\sum x_i^2}{N} - \frac{\sum x_i y_i}{N} = 0 \end{cases}$$

$$\mathrm{Var}(x) = \frac{\sum x_i^2}{N} - \bar{x}^2.$$

$$\mathrm{covar}(x, y) = \frac{\sum x_i y_i}{N} - \bar{x} \cdot \bar{y}$$

$$\begin{cases} w_0 + w_1 \cdot \bar{x} - \bar{y} = 0 & \Rightarrow \quad w_0 = \bar{y} - w_1 \cdot \bar{x} \\ w_0 \cdot \bar{x} + w_1 \cdot \frac{\sum x_i^2}{N} - \frac{\sum x_i y_i}{N} = 0 & \end{cases}$$

$$\bar{y} \cdot \bar{x} - \underbrace{w_1 \bar{x}^2 + w_1 \cdot \frac{\sum x_i^2}{N}}_{w_1 \cdot \operatorname{var}(x)} - \frac{\sum x_i y_i}{N} = 0,$$

$$w_1 \cdot \operatorname{var}(x) = \operatorname{covar}(x, y)$$

$$\begin{cases} w_1 = \dfrac{\operatorname{covar}(x,y)}{\operatorname{var}(x)} \\ w_0 = \bar{y} - w_1 \cdot \bar{x}. \end{cases}$$



## Computing the Model Parameters: Summary

$\langle (x_i, y_i) \rangle_{i=1}^{N}$, where $x_i, y_i \in \mathbb{R}$ $\qquad \widehat{y}(x) = w_0 + x \cdot w_1$ (no noise term)

$$\mathcal{L}(\mathbf{w}) = \mathcal{L}(w_0, w_1) = \frac{1}{2N} \sum_{i=1}^{N} (\widehat{y}_i - y_i)^2 = \frac{1}{2N} \sum_{i=1}^{N} (w_0 + x_i \cdot w_1 - y_i)^2$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{1}{N} \sum_{i=1}^{N} (w_0 + w_1 \cdot x_i - y_i)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{1}{N} \sum_{i=1}^{N} (w_0 + w_1 \cdot x_i - y_i) x_i$$

We obtain the solution for $(w_0, w_1)$ by setting the partial derivatives to 0 and solving the resulting system. (Normal Equations)

$$w_0 + w_1 \cdot \frac{\sum_i x_i}{N} = \frac{\sum_i y_i}{N} \qquad (1)$$

$$w_0 \cdot \frac{\sum_i x_i}{N} + w_1 \cdot \frac{\sum_i x_i^2}{N} = \frac{\sum_i x_i y_i}{N} \qquad (2)$$

$$\boxed{\begin{aligned} \bar{x} &= \frac{\sum_i x_i}{N} \\ \bar{y} &= \frac{\sum_i y_i}{N} \\ \widehat{\operatorname{var}}(x) &= \frac{\sum_i x_i^2}{N} - \bar{x}^2 \\ \widehat{\operatorname{cov}}(x, y) &= \frac{\sum_i x_i y_i}{N} - \bar{x} \cdot \bar{y} \\ \\ w_1 &= \frac{\widehat{\operatorname{cov}}(x, y)}{\widehat{\operatorname{var}}(x)} \\ w_0 &= \bar{y} - w_1 \cdot \bar{x} \end{aligned}}$$



## Linear Regression : General Case

Recall that the linear model is

$$\widehat{y}_i = \sum_{j=0}^{D} x_{ij} w_j$$

where we assume that $x_{i0} = 1$ for all $\mathbf{x}_i$, so that the bias term $w_0$ does not need to be treated separately.

Expressing everything in matrix notation

$$\widehat{\mathbf{y}} = \mathbf{X}\mathbf{w}$$

Here we have $\widehat{\mathbf{y}} \in \mathbb{R}^{N \times 1}$, $\mathbf{X} \in \mathbb{R}^{N \times (D+1)}$ and $\mathbf{w} \in \mathbb{R}^{(D+1) \times 1}$

$$\underset{\widehat{\mathbf{y}}_{N\times 1}}{\begin{bmatrix} \widehat{y}_1 \\ \widehat{y}_2 \\ \vdots \\ \widehat{y}_N \end{bmatrix}} = \underset{\mathbf{X}_{N\times(D+1)}}{\begin{bmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \\ \vdots \\ \mathbf{x}_N^{\mathsf{T}} \end{bmatrix}} \underset{\mathbf{w}_{(D+1)\times 1}}{\begin{bmatrix} w_0 \\ \vdots \\ w_D \end{bmatrix}} = \underset{\mathbf{X}_{N\times(D+1)}}{\begin{bmatrix} x_{10} & \cdots & x_{1D} \\ x_{20} & \cdots & x_{2D} \\ \vdots & \ddots & \vdots \\ x_{N0} & \cdots & x_{ND} \end{bmatrix}} \underset{\mathbf{w}_{(D+1)\times 1}}{\begin{bmatrix} w_0 \\ \vdots \\ w_D \end{bmatrix}}$$



## Back to Toy Example

| one | dist (km) | weekday? | commute time (min) |
|---|---|---|---|
| 1 | 2.7 | 1 (fri) | 25 |
| 1 | 4.1 | 1 (mon) | 33 |
| 1 | 1.0 | 0 (sun) | 15 |
| 1 | 5.2 | 1 (tue) | 45 |
| 1 | 2.8 | 0 (sat) | 22 |

We have $N = 5$, $D + 1 = 3$ and so we get

$$\mathbf{y} = \begin{bmatrix} 25 \\ 33 \\ 15 \\ 45 \\ 22 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & 2.7 & 1 \\ 1 & 4.1 & 1 \\ 1 & 1.0 & 0 \\ 1 & 5.2 & 1 \\ 1 & 2.8 & 0 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

For $\mathbf{w} = [6.09, 6.53, 2.11]^{\mathsf{T}}$, our predictions would be $\widehat{\mathbf{y}} = \begin{bmatrix} 25.83 \\ 34.97 \\ 12.62 \\ 42.16 \\ 24.37 \end{bmatrix}$



## Finding Optimal Solutions using Calculus

$$\begin{aligned} \mathcal{L}(\mathbf{w}) &= \frac{1}{2N} \sum_{i=1}^{N} (\mathbf{x}_i^{\mathsf{T}} \mathbf{w} - y_i)^2 = \frac{1}{2N} (\mathbf{X}\mathbf{w} - \mathbf{y})^{\mathsf{T}} (\mathbf{X}\mathbf{w} - \mathbf{y}) \\ &= \frac{1}{2N} \left( \mathbf{w}^{\mathsf{T}} \left( \mathbf{X}^{\mathsf{T}} \mathbf{X} \right) \mathbf{w} - \mathbf{w}^{\mathsf{T}} \mathbf{X}^{\mathsf{T}} \mathbf{y} - \mathbf{y}^{\mathsf{T}} \mathbf{X} \mathbf{w} + \mathbf{y}^{\mathsf{T}} \mathbf{y} \right) \\ &= \frac{1}{2N} \left( \mathbf{w}^{\mathsf{T}} \left( \mathbf{X}^{\mathsf{T}} \mathbf{X} \right) \mathbf{w} - 2 \cdot \mathbf{y}^{\mathsf{T}} \mathbf{X} \mathbf{w} + \mathbf{y}^{\mathsf{T}} \mathbf{y} \right) \\ &= \cdots \end{aligned}$$

Then, write out all partial derivatives to form the gradient $\nabla_{\mathbf{w}} \mathcal{L}$

$$\frac{\partial \mathcal{L}}{\partial w_0} = \cdots$$

$$\frac{\partial \mathcal{L}}{\partial w_1} = \cdots$$

$$\vdots$$

$$\frac{\partial \mathcal{L}}{\partial w_D} = \cdots$$

Instead, we will use matrix calculus shortcuts to differentiate using matrix notation directly



## Differentiating Matrix Expressions

Rules (Tricks)

(i) Linear Form Expressions: $\nabla_{\mathbf{w}} \left( \mathbf{c}^{\mathsf{T}} \mathbf{w} \right) = \mathbf{c}$

$$\mathbf{c}^{\mathsf{T}} \mathbf{w} = \sum_{j=0}^{D} c_j w_j$$

$$\frac{\partial \left( \mathbf{c}^{\mathsf{T}} \mathbf{w} \right)}{\partial w_j} = c_j, \qquad \text{and so} \quad \nabla_{\mathbf{w}} \left( \mathbf{c}^{\mathsf{T}} \mathbf{w} \right) = \mathbf{c} \qquad (3)$$

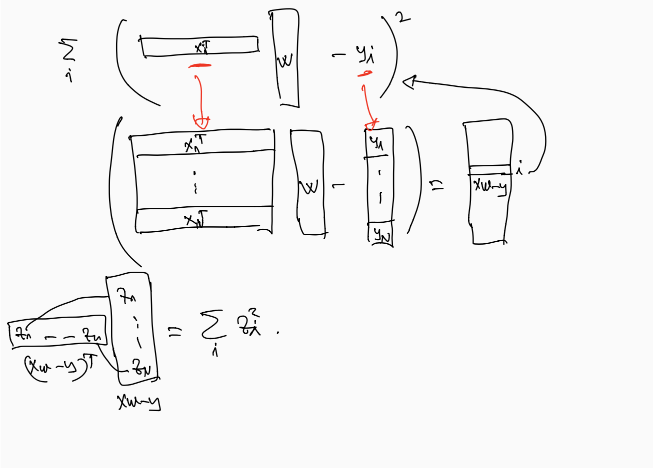
(ii) Quadratic Form Expressions:

$$\nabla_{\mathbf{w}} \left( \mathbf{w}^{\mathsf{T}} \mathbf{A} \mathbf{w} \right) = \mathbf{A}\mathbf{w} + \mathbf{A}^{\mathsf{T}} \mathbf{w} \quad (= 2\mathbf{A}\mathbf{w} \text{ for symmetric } \mathbf{A})$$

$$\mathbf{w}^{\mathsf{T}} \mathbf{A} \mathbf{w} = \sum_{i=0}^{D} \sum_{j=0}^{D} w_i w_j A_{ij}$$

$$\frac{\partial \left( \mathbf{w}^{\mathsf{T}} \mathbf{A} \mathbf{w} \right)}{\partial w_k} = \sum_{i=0}^{D} w_i A_{ik} + \sum_{j=0}^{D} A_{kj} w_j = \mathbf{A}_{[:,k]}^{\mathsf{T}} \mathbf{w} + \mathbf{A}_{[k,:]} \mathbf{w}$$

$$\nabla_{\mathbf{w}} \left( \mathbf{w}^{\mathsf{T}} \mathbf{A} \mathbf{w} \right) = \mathbf{A}^{\mathsf{T}} \mathbf{w} + \mathbf{A} \mathbf{w} \qquad (4)$$




18

$$\mathcal{L}(w) = \frac{1}{2N} \underbrace{(Xw - y)^T}_{} (Xw - y)$$

$$(w^T X^T - y^T)(Xw - y)$$

$$w^T \underbrace{X^T X}_{} w - \underbrace{w^T X^T y - y^T X w}_{y^T X w} + \underbrace{y^T y}_{}$$

$$\underbrace{w^T X^T y}_{1 \times (D+1)\;\; (D+1) \times N\;\; N \times 1} = \square_{1 \times 1}$$

$$\mathcal{L}(w) = \left(w^T (X^T X) w - 2 y^T X w + y^T y\right) \frac{1}{2N}$$

$$-2 y^T X w.$$

$$\nabla_w \mathcal{L} = \frac{1}{N} \cdot \left( X^T X w - X^T y \right) = 0$$

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

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

Predictions on $X$: $\hat{y} = X w = \boxed{X (X^T X)^{-1} X^T} y$

hat matrix

19

## Deriving the Least Squares Estimate: Summary

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2N} \sum_{i=1}^{N} (\mathbf{x}_i^{\mathsf{T}} \mathbf{w} - y_i)^2 = \frac{1}{2N} \left( \mathbf{w}^{\mathsf{T}} \left( \mathbf{X}^{\mathsf{T}} \mathbf{X} \right) \mathbf{w} - 2 \cdot \mathbf{y}^{\mathsf{T}} \mathbf{X} \mathbf{w} + \mathbf{y}^{\mathsf{T}} \mathbf{y} \right)$$

We compute the gradient $\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{0}$ using the matrix differentiation rules,

$$\nabla_{\mathbf{w}} \mathcal{L} = \frac{1}{N} \left( \left( \mathbf{X}^{\mathsf{T}} \mathbf{X} \right) \mathbf{w} - \mathbf{X}^{\mathsf{T}} \mathbf{y} \right)$$

By setting $\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{0}$ and solving we get,

$$\left( \mathbf{X}^{\mathsf{T}} \mathbf{X} \right) \mathbf{w} = \mathbf{X}^{\mathsf{T}} \mathbf{y}$$

$$\mathbf{w} = \left( \mathbf{X}^{\mathsf{T}} \mathbf{X} \right)^{-1} \mathbf{X}^{\mathsf{T}} \mathbf{y} \qquad \text{(Assuming inverse exists)}$$

The predictions made by the model on the data $\mathbf{X}$ are given by

$$\widehat{\mathbf{y}} = \mathbf{X} \mathbf{w} = \mathbf{X} \left( \mathbf{X}^{\mathsf{T}} \mathbf{X} \right)^{-1} \mathbf{X}^{\mathsf{T}} \mathbf{y}$$

$\mathbf{X} \left( \mathbf{X}^{\mathsf{T}} \mathbf{X} \right)^{-1} \mathbf{X}^{\mathsf{T}}$ is called the "hat" matrix

20

## Complexity of Parameter Estimation

$$\mathbf{w} = \underbrace{\underbrace{\left( \underbrace{\underbrace{\mathbf{X}^{\mathsf{T}}}_{(D+1) \times N} \underbrace{\mathbf{X}}_{N \times (D+1)}}_{(D+1) \times (D+1)} \right)^{-1}}_{(D+1) \times (D+1)} \underbrace{\mathbf{X}^{\mathsf{T}}}_{(D+1) \times N} \underbrace{\mathbf{y}}_{N \times 1}}_{(D+1) \times 1}$$

- $\mathbf{Z} = \mathbf{X}^{\mathsf{T}} \mathbf{X}$ in $O(D^2 N)$
  - If $D = O(N)$, then the best known method (Le Gall) needs $O(N^{2.37})$
- $\mathbf{Z}^{-1}$ in $O(D^3)$
- $\mathbf{A} = \mathbf{X}^{\mathsf{T}} \mathbf{y}$ in $O(DN)$
- $\mathbf{w} = \mathbf{Z}^{-1} \mathbf{A}$ in $O(D^2)$

Overall complexity for computing $\mathbf{w}$: $O(D^2 \mathbf{N} + D^3)$

21

## Complexity of Parameter Estimation

What if $\mathbf{X}$ is defined by a join of several relations?

- The number of rows $N$ may be exponential in the number of relations:

$$N = O(M^{\text{number relations}})$$

- $\mathbf{X}$ is sparse, it can be represented in $O(M)$ space losslessly for acyclic joins

  Acyclic joins are common in practice

- $\mathbf{w}$ can be computed in $O(D^2 \mathbf{M} + D^3)$

- Find out more: `https://fdbresearch.github.io/`

22

## When Do We Expect $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ to be Invertible?

Matrix $(\mathbf{X}^{\mathsf{T}} \mathbf{X}) \in \mathbb{R}^{(D+1) \times (D+1)}$

- $\mathrm{rank}(\mathbf{X}^{\mathsf{T}} \mathbf{X}) = \mathrm{rank}(\mathbf{X}) \leq \min\{D + 1, N\}$
- It is invertible if $\mathrm{rank}(\mathbf{X}) = D + 1$

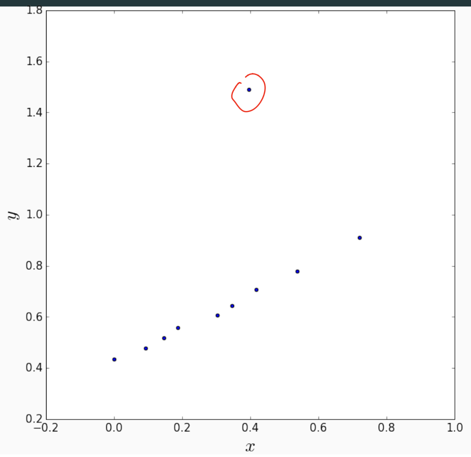
What if we use one-hot encoding for a feature like day?

- $x_{\mathrm{mon}}, \ldots, x_{\mathrm{sun}}$ stand for 0-1 valued variables in the one-hot encoding
- We always have $x_{\mathrm{mon}} + \cdots + x_{\mathrm{sun}} = 1$
- This introduces a linear dependence in the columns of $\mathbf{X}$ reducing the rank
- In this case, we can drop some features to adjust rank

  We'll see alternative approaches later in the course

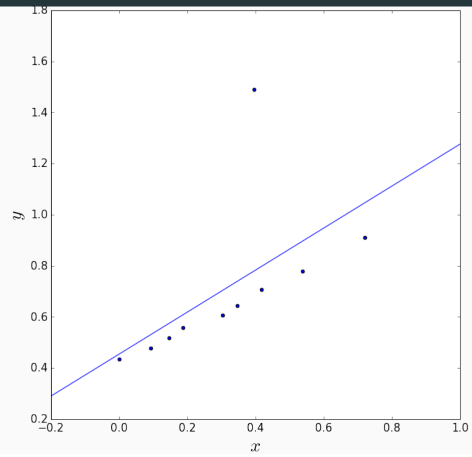
23

## Least Squares Estimate in the Presence of Outlliers



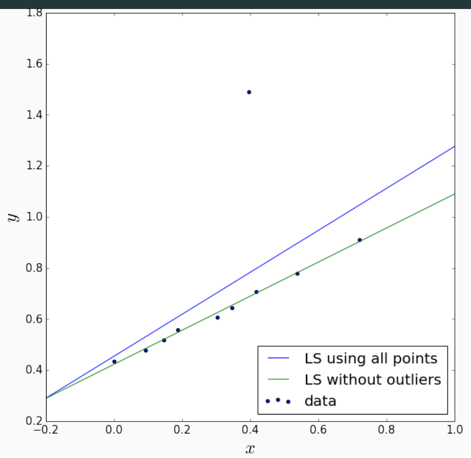



## Least Squares Estimate in the Presence of Outlliers






## Least Squares Estimate in the Presence of Outlliers






## Least Squares Estimate in the Presence of Outlliers