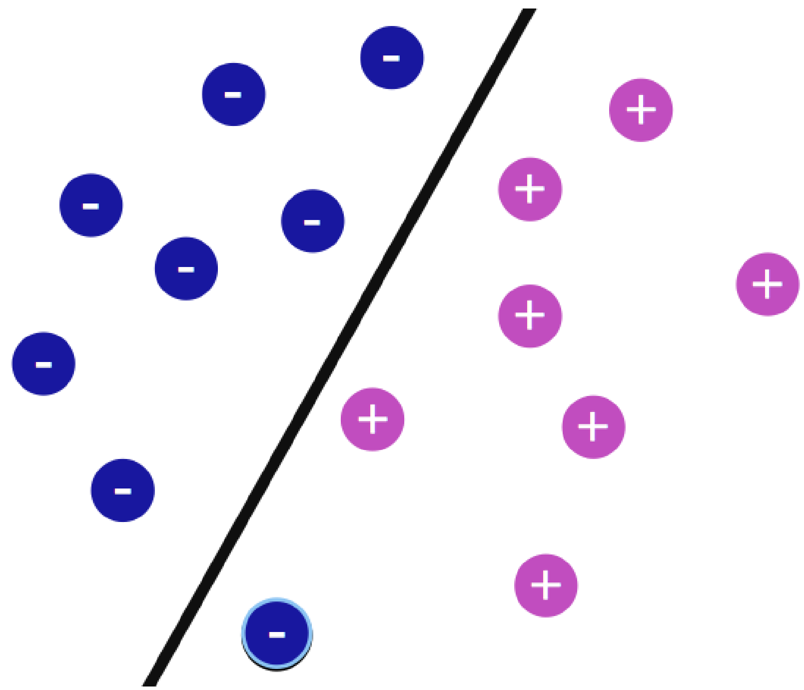
Machine Learning

Linear Models



Linear Regression: History

- A very popular technique.
- Rooted in Statistics.
- Method of Least Squares used as early as 1795 by Gauss.
- Re-invented in 1805 by Legendre.
- Frequently applied in **astronomy** to study the large scale of the universe.
- Still a very useful tool today.

Carl Friedrich Gauss

Linear Regression

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n)$ / $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$

Linear Regression

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n)$ / $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$

example $x_1 \rightarrow$	x_{11}	x_{12}	$\ldots$	x_{1d}	$y_1 \leftarrow$ label
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
example $x_i \rightarrow$	x_{i1}	x_{i2}	$\ldots$	x_{id}	$y_i \leftarrow$ label
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
example $x_n \rightarrow$	x_{n1}	x_{n2}	$\ldots$	x_{nd}	$y_n \leftarrow$ label

Linear Regression

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n)$ / $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$

example $x_1 \rightarrow$	x_{11}	x_{12}	$\ldots$	x_{1d}	$y_1 \leftarrow$ label
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
example $x_i \rightarrow$	x_{i1}	x_{i2}	$\ldots$	x_{id}	$y_i \leftarrow$ label
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
example $x_n \rightarrow$	x_{n1}	x_{n2}	$\ldots$	x_{nd}	$y_n \leftarrow$ label

Task: Learn a regression function:

$$f : \mathbb{R}^d \longrightarrow \mathbb{R}$$

$$f(x) = y$$

Linear Regression

Given: Training data: $(x_1, y_1), \ldots, (x_n, y_n)$ / $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$

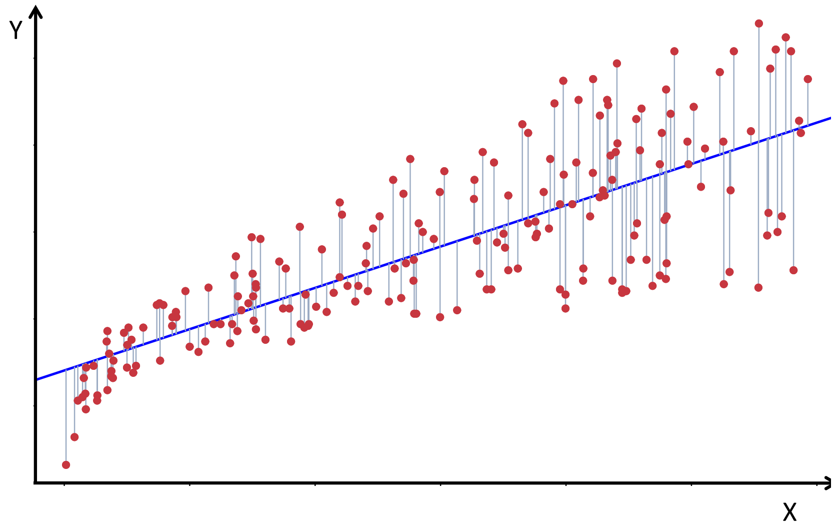
example $x_1 \rightarrow$	x_{11}	x_{12}	$\ldots$	x_{1d}	$y_1 \leftarrow$ label
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
example $x_i \rightarrow$	x_{i1}	x_{i2}	$\ldots$	x_{id}	$y_i \leftarrow$ label
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
example $x_n \rightarrow$	x_{n1}	x_{n2}	$\ldots$	x_{nd}	$y_n \leftarrow$ label

Task: Learn a regression function:

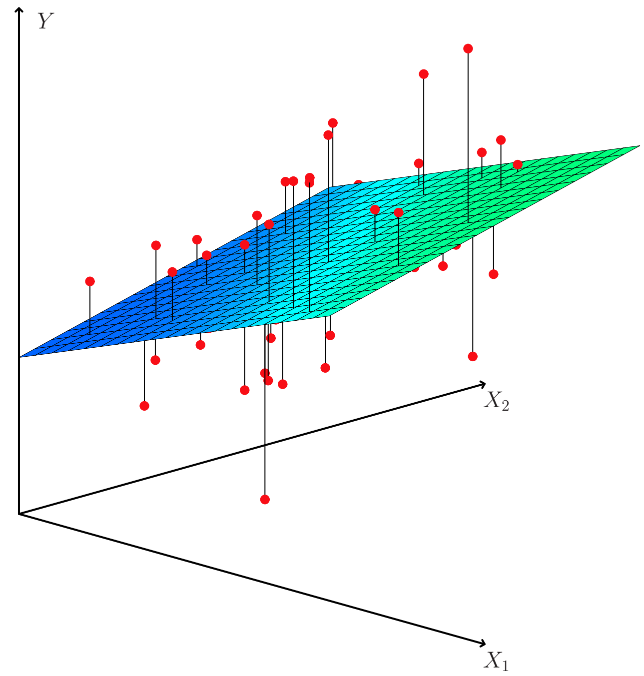
$$f : \mathbb{R}^d \longrightarrow \mathbb{R}$$
$$f(x) = y$$

Linear Regression: A regression model is said to be linear if it is represented by a linear function.

Linear Regression



$d = 1$, line in $\mathbb{R}^2$



$d = 2$, hyperplane is $\mathbb{R}^3$

Credit: Introduction to Statistical Learning.

Linear Regression

Linear Regression Model:

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j \quad \text{with} \quad \beta_j \in \mathbb{R}, \quad j \in \{1, \ldots, d\}$$

β 's are called parameters or coefficients or weights.

Linear Regression

Linear Regression Model:

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j \quad \text{with} \quad \beta_j \in \mathbb{R}, \quad j \in \{1, \ldots, d\}$$

β 's are called parameters or coefficients or weights.

$$\boxed{\text{Learning the linear model} \longrightarrow \text{learning the } \beta's}$$

Linear Regression

Linear Regression Model:

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j \quad \text{with} \quad \beta_j \in \mathbb{R}, \quad j \in \{1, \ldots, d\}$$

β 's are called parameters or coefficients or weights.

Learning the linear model $\longrightarrow$ learning the $\beta' s$

Estimation with Least squares:

Use least square loss: $\ell oss(y_i, f(x_i)) = (y_i - f(x_i))^2$

Linear Regression

Linear Regression Model:

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j \quad \text{with} \quad \beta_j \in \mathbb{R}, \quad j \in \{1, \ldots, d\}$$

β 's are called parameters or coefficients or weights.

$\boxed{\text{Learning the linear model} \longrightarrow \text{learning the } \beta' s}$

Estimation with Least squares:

Use least square loss: $\ell oss(y_i, f(x_i)) = (y_i - f(x_i))^2$

We want to minimize the loss over all examples, that is minimize the *risk or cost function* R:

$$R = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Linear Regression

A simple case with one feature ($d = 1$):

$$f(x) = \beta_0 + \beta_1 x$$

Linear Regression

A simple case with one feature ($d = 1$):

$$f(x) = \beta_0 + \beta_1 x$$

We want to minimize:

$$R = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

Linear Regression

A simple case with one feature ($d = 1$):

$$f(x) = \beta_0 + \beta_1 x$$

We want to minimize:

$$R = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$R(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Linear Regression

A simple case with one feature ($d = 1$):

$$f(x) = \beta_0 + \beta_1 x$$

We want to minimize:

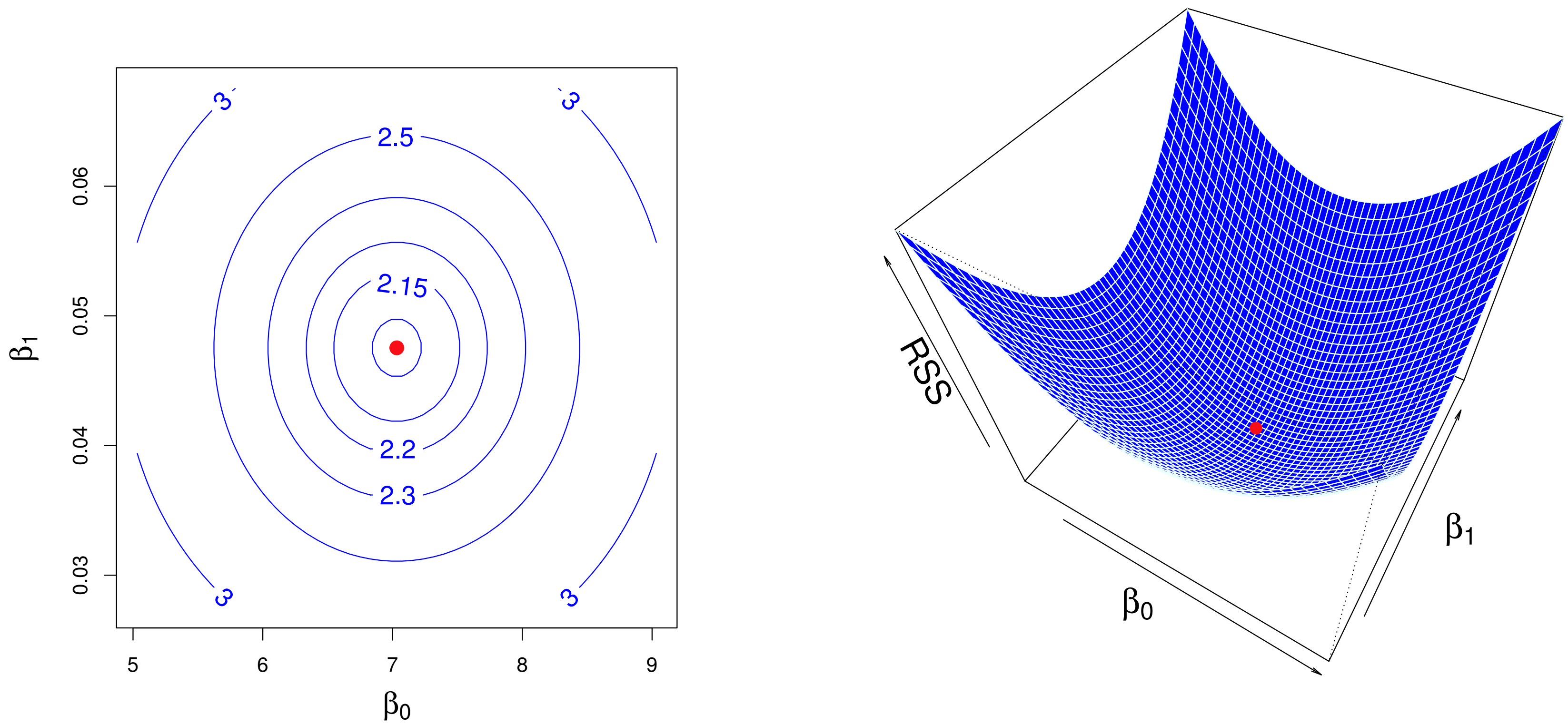
$$R = \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$R(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Find β_0 and β_1 that minimize:

$$R(\beta) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Linear Regression



Credit: Introduction to Statistical Learning.

Linear Regression

Find β_0 and β_1 so that:

$$argmin_{\beta}(\frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2)$$

Linear Regression

Find β_0 and β_1 so that:

$$argmin_{\beta}(\frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2)$$

Minimize: $R(\beta_0, \beta_1)$, that is: $\frac{\partial R}{\partial \beta_0} = 0 \qquad \frac{\partial R}{\partial \beta_1} = 0$

Linear Regression

Find β_0 and β_1 so that:

$$argmin_{\beta}(\frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2)$$

Minimize: $R(\beta_0, \beta_1)$, that is: $\frac{\partial R}{\partial \beta_0} = 0 \qquad \frac{\partial R}{\partial \beta_1} = 0$

$$\frac{\partial R}{\partial \beta_0} = 2 \times \frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \times \frac{\partial}{\partial \beta_0}(y_i - \beta_0 - \beta_1 x_i)$$

Linear Regression

Find β_0 and β_1 so that:

$$argmin_\beta(\frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2)$$

Minimize: $R(\beta_0, \beta_1)$, that is: $\frac{\partial R}{\partial \beta_0} = 0 \qquad \frac{\partial R}{\partial \beta_1} = 0$

$$\frac{\partial R}{\partial \beta_0} = 2 \times \frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \times \frac{\partial}{\partial \beta_0}(y_i - \beta_0 - \beta_1 x_i)$$

$$\frac{\partial R}{\partial \beta_0} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \times (-1) = 0$$

Linear Regression

Find β_0 and β_1 so that:

$$argmin_\beta\left(\frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2\right)$$

Minimize: $R(\beta_0, \beta_1)$, that is: $\frac{\partial R}{\partial \beta_0} = 0 \qquad \frac{\partial R}{\partial \beta_1} = 0$

$$\frac{\partial R}{\partial \beta_0} = 2 \times \frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \times \frac{\partial}{\partial \beta_0}(y_i - \beta_0 - \beta_1 x_i)$$

$$\frac{\partial R}{\partial \beta_0} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \times (-1) = 0$$

$$\beta_0 = \frac{1}{n}\sum_{i=1}^{n} y_i - \beta_1 \frac{1}{n}\sum_{i=1}^{n} x_i$$

Linear Regression

$$\frac{\partial R}{\partial \beta_1} = 2 \times \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times \frac{\partial}{\partial \beta_1} (y_i - \beta_0 - \beta_1 x_i)$$

Linear Regression

$$\frac{\partial R}{\partial \beta_1} = 2 \times \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times \frac{\partial}{\partial \beta_1} (y_i - \beta_0 - \beta_1 x_i)$$

$$\frac{\partial R}{\partial \beta_1} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times (-x_i) = 0$$

Linear Regression

$$\frac{\partial R}{\partial \beta_1} = 2 \times \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times \frac{\partial}{\partial \beta_1} (y_i - \beta_0 - \beta_1 x_i)$$

$$\frac{\partial R}{\partial \beta_1} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times (-x_i) = 0$$

$$\beta_1 \sum_{i=1}^{n} {x_i}^2 = \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} \beta_0 x_i$$

Linear Regression

$$\frac{\partial R}{\partial \beta_1} = 2 \times \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times \frac{\partial}{\partial \beta_1} (y_i - \beta_0 - \beta_1 x_i)$$

$$\frac{\partial R}{\partial \beta_1} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times (-x_i) = 0$$

$$\beta_1 \sum_{i=1}^{n} {x_i}^2 = \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} \beta_0 x_i$$

Plugging β_0 in β_1:

$$\beta_1 = \frac{\sum_{i=1}^{n} y_i x_i - \frac{1}{n} \sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n} \sum_{i=1}^{n} x_i \sum x_i}$$

Linear Regression

With more than one feature:

$$f(x) = \beta_0 + \sum_{j=1}^{d} \beta_j x_j$$

Find the β_j that minimize:

$$R = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{d} \beta_j x_{ij}))^2$$

Let's write it more elegantly with matrices!

Matrix representation

Let X be an $n \times (d+1)$ matrix where each row starts with a 1 followed by a feature vector.
Let y be the label vector of the training set.
Let β be the vector of weights (that we want to estimate!).

$$X := \begin{pmatrix} 1 & x_{11} & \cdots & x_{1j} & \cdots & x_{1d} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{i1} & \cdots & x_{ij} & \cdots & x_{id} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nj} & \cdots & x_{nd} \end{pmatrix}$$

$$y := \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix} \qquad \beta := \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_j \\ \vdots \\ \beta_d \end{pmatrix}$$

Normal Equation

We want to find $(d+1)$ β's that minimize R. We write R:

$$R(\beta) = \frac{1}{2n}||(y - X\beta)||^2$$

$$R(\beta) = \frac{1}{2n}(y - X\beta)^T(y - X\beta)$$

$$\frac{\partial R}{\partial \beta} = -\frac{1}{n}X^T(y - X\beta)$$

We have that:

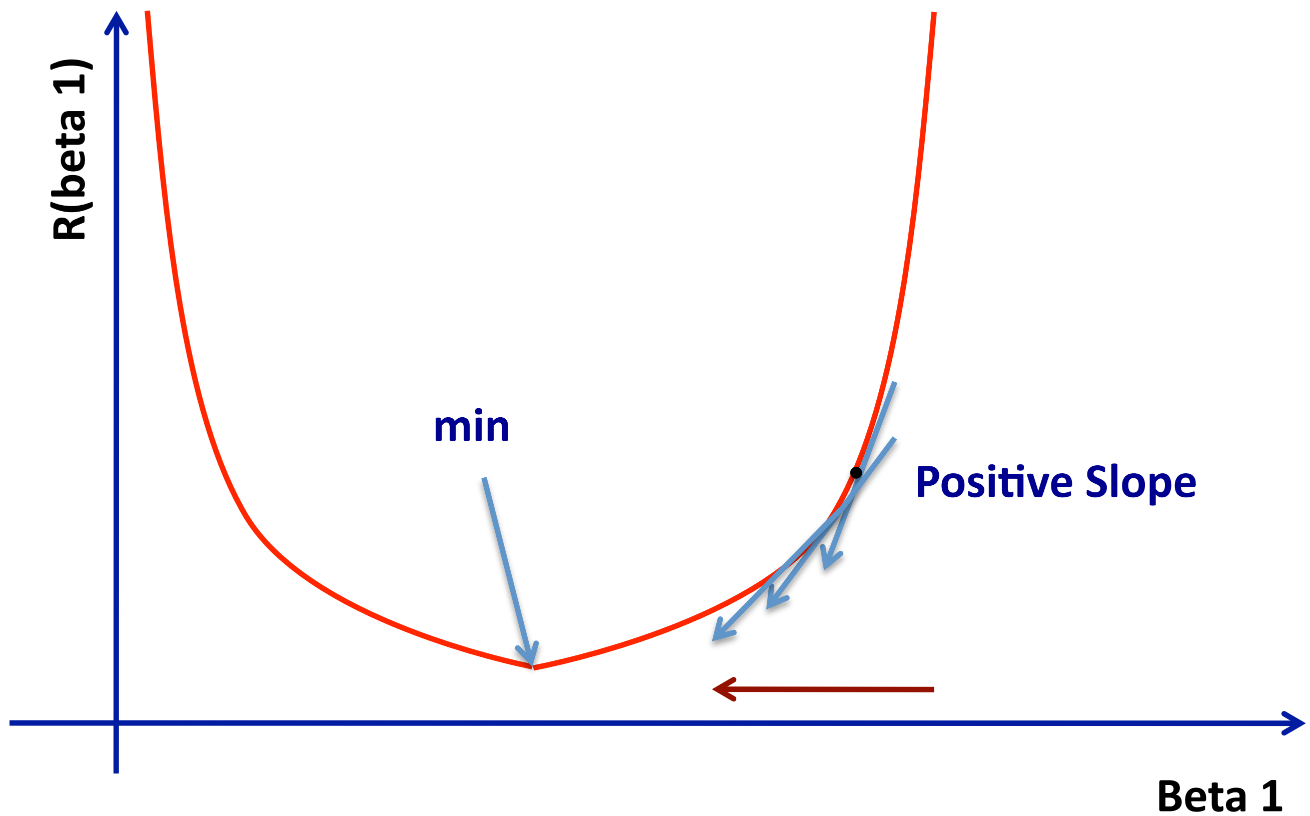
$$\frac{\partial^2 R}{\partial \beta} = -\frac{1}{n}X^T X$$

is positive definite which ensures that β is a minimum. We solve:

$$X^T(y - X\beta) = 0$$

The unique solution is: $\boxed{\beta = (X^T X)^{-1} X^T y}$

Gradient descent



Gradient descent

Gradient Descent is an optimization method.

Repeat until convergence:

Update **simultaneously** all β_j for ($j = 0$ and $j = 1$)

$$\beta_0 := \beta_0 - \alpha \frac{\partial}{\partial \beta_0} R(\beta_0, \beta_1)$$

$$\beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} R(\beta_0, \beta_1)$$

Gradient descent

Gradient Descent is an optimization method.

Repeat until convergence:

Update **simultaneously** all β_j for ($j = 0$ and $j = 1$)

$$\beta_0 := \beta_0 - \alpha \frac{\partial}{\partial \beta_0} R(\beta_0, \beta_1)$$

$$\beta_1 := \beta_1 - \alpha \frac{\partial}{\partial \beta_1} R(\beta_0, \beta_1)$$

α is a learning rate.

Gradient descent

In the linear case:

$$\frac{\partial R}{\partial \beta_0} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times (-1) = 0$$

$$\frac{\partial R}{\partial \beta_1} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) \times (-x_i)$$

Let's generalize it!

Gradient descent

In the linear case:

$$\frac{\partial R}{\partial \beta_0} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \times (-1) = 0$$

$$\frac{\partial R}{\partial \beta_1} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i) \times (-x_i)$$

Repeat until convergence:

Update **simultaneously** all β_j for ($j = 0$ and $j = 1$)

$$\beta_0 := \beta_0 - \alpha \frac{1}{n}\sum_{i=1}^{n}(\beta_0 + \beta_1 x_i - y_i)$$

$$\beta_1 := \beta_1 - \alpha \frac{1}{n}\sum_{i=1}^{n}(\beta_0 + \beta_1 x_i - y_i)(x_i)$$

Pros and Cons

Analytical approach: Normal Equation

+ No need to specify a convergence rate or iterate.
- Works only if $X^T X$ is invertible
- Very slow if d is large $O(d^3)$ to compute $(X^T X)^{-1}$

Iterative approach: Gradient Descent

+ Effective and efficient even in high dimensions.
- Iterative (sometimes need many iterations to converge).
- Needs to choose the rate α.

Practical considerations

1. **Scaling**: Bring your features to a similar scale.

Practical considerations

1. **Scaling**: Bring your features to a similar scale.

$$x_i := \frac{x_i - \mu_i}{stdev(x_i)}$$

Practical considerations

1. **Scaling**: Bring your features to a similar scale.

$$x_i := \frac{x_i - \mu_i}{stdev(x_i)}$$

2. **Learning rate**: Don't use a rate that is too small or too large.

Practical considerations

1. **Scaling**: Bring your features to a similar scale.

$$x_i := \frac{x_i - \mu_i}{stdev(x_i)}$$

2. **Learning rate**: Don't use a rate that is too small or too large.

3. **R should decrease** after each iteration.

Practical considerations

1. **Scaling**: Bring your features to a similar scale.

$$x_i := \frac{x_i - \mu_i}{stdev(x_i)}$$

2. **Learning rate**: Don't use a rate that is too small or too large.

3. **R should decrease** after each iteration.

4. **Declare convergence** if it start decreasing by less ϵ

Practical considerations

1. **Scaling**: Bring your features to a similar scale.

$$x_i := \frac{x_i - \mu_i}{stdev(x_i)}$$

2. **Learning rate**: Don't use a rate that is too small or too large.

3. **R should decrease** after each iteration.

4. **Declare convergence** if it start decreasing by less ϵ

5. If $X^T X$ is not **invertible**?

Practical considerations

1. **Scaling**: Bring your features to a similar scale.

$$x_i := \frac{x_i - \mu_i}{stdev(x_i)}$$

2. **Learning rate**: Don't use a rate that is too small or too large.

3. **R should decrease** after each iteration.

4. **Declare convergence** if it start decreasing by less ϵ

5. If $X^T X$ is not **invertible**?
   (a) Too many features as compared to the number of examples (e.g., 50 examples and 500 features)
   (b) Features linearly dependent: e.g., weight in pounds and in kilo.

Credit

- The elements of statistical learning. Data mining, inference, and prediction. 10th Edition 2009. T. Hastie, R. Tibshirani, J. Friedman.
- Machine Learning 1997. Tom Mitchell.