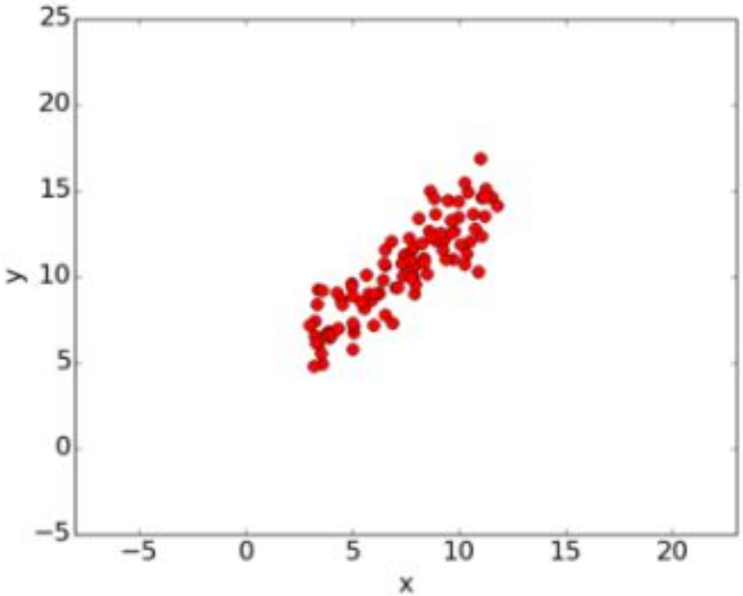
Linear Regression – Numerical Solution

Iterative solution

Linear regression – an example that uses gradient descent



We want to fit a straight line (in 2D case). The sample we are given with is $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$.

We believe the relation between $x\ and\ y$:

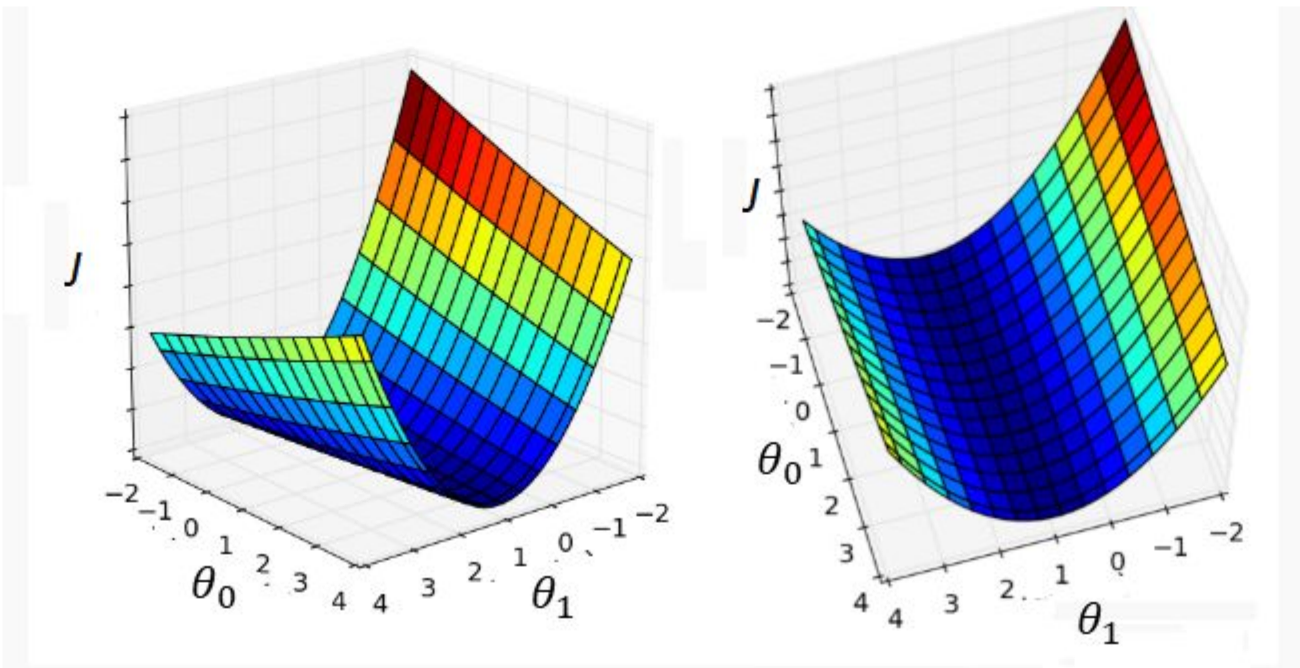
$$y = \theta_0 + \theta_1 x = h(x)$$

We want to find (θ_0, θ_1).

In the space (θ_0, θ_1). We want to find the best solution.

Sum of Squared Error = J(θ_0 , θ_1)

J(θ_0 , θ_1) $= \sum_{i=1}^{n}(y_i - (\theta_0 + \theta_1 x_i))^2$



$$\nabla_{\binom{\theta_0}{\theta_1}} J = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \frac{\partial J}{\partial \theta_1} \end{bmatrix}$$

$$\frac{\partial J}{\partial \theta_0} = 2\sum_{i=1}^{n} -(y_i - (\theta_0 + \theta_1 x_i))$$

$$\frac{\partial J}{\partial \theta_1} = 2\sum_{i=1}^{n} -x_i(y_i - (\theta_0 + \theta_1 x_i))$$

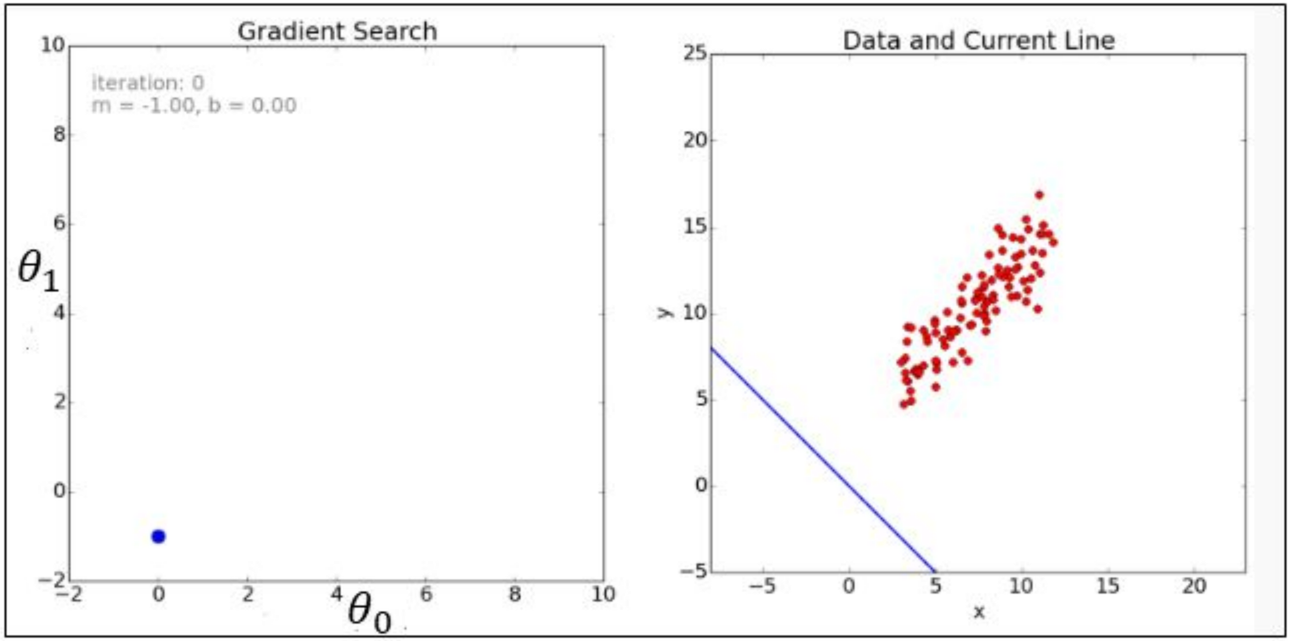
Gradient Descent Method:

Begin with random $\Theta = \binom{\theta_0}{\theta_1}$, we call this Θ_0

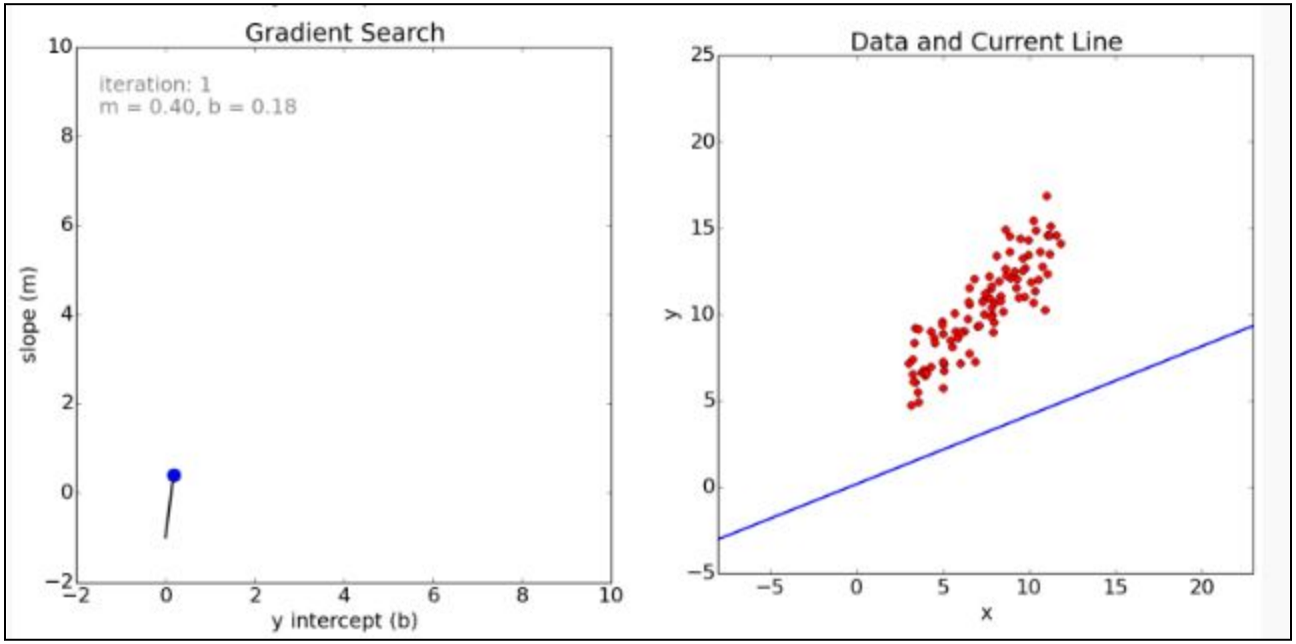
$\Theta_1 = \Theta_0 - \eta \nabla \mathrm{J}$

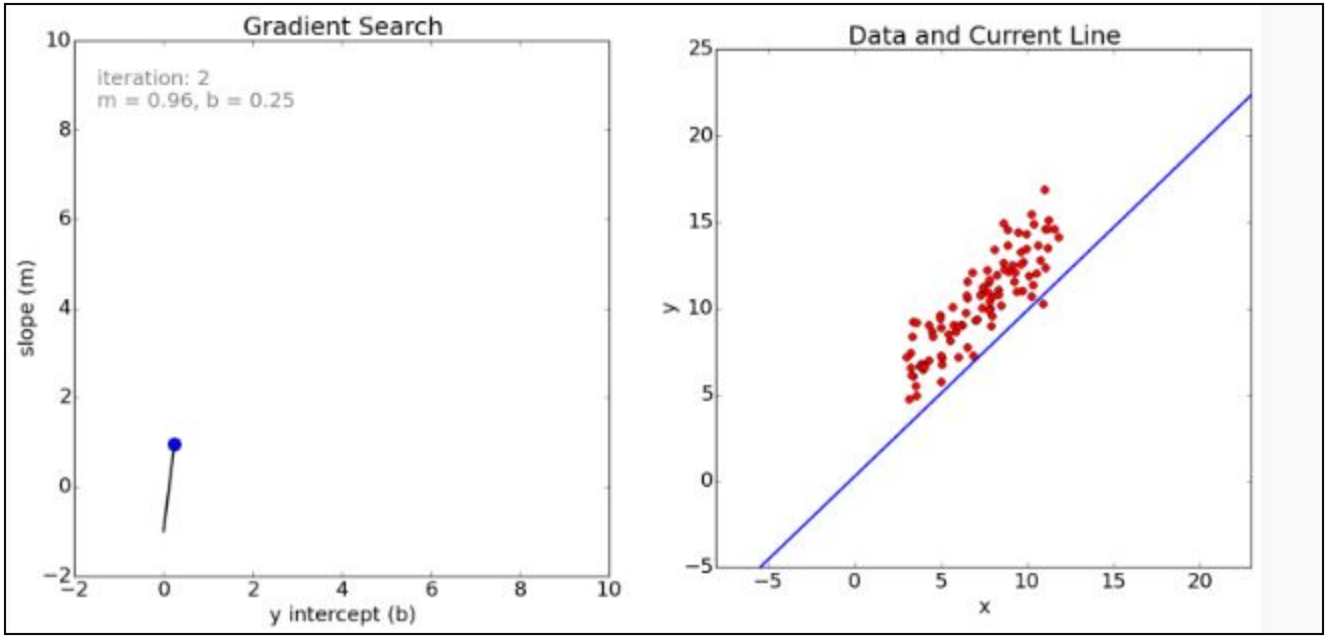
In general $\Theta_{k+1} = \Theta_k - \eta \nabla \mathrm{J}$

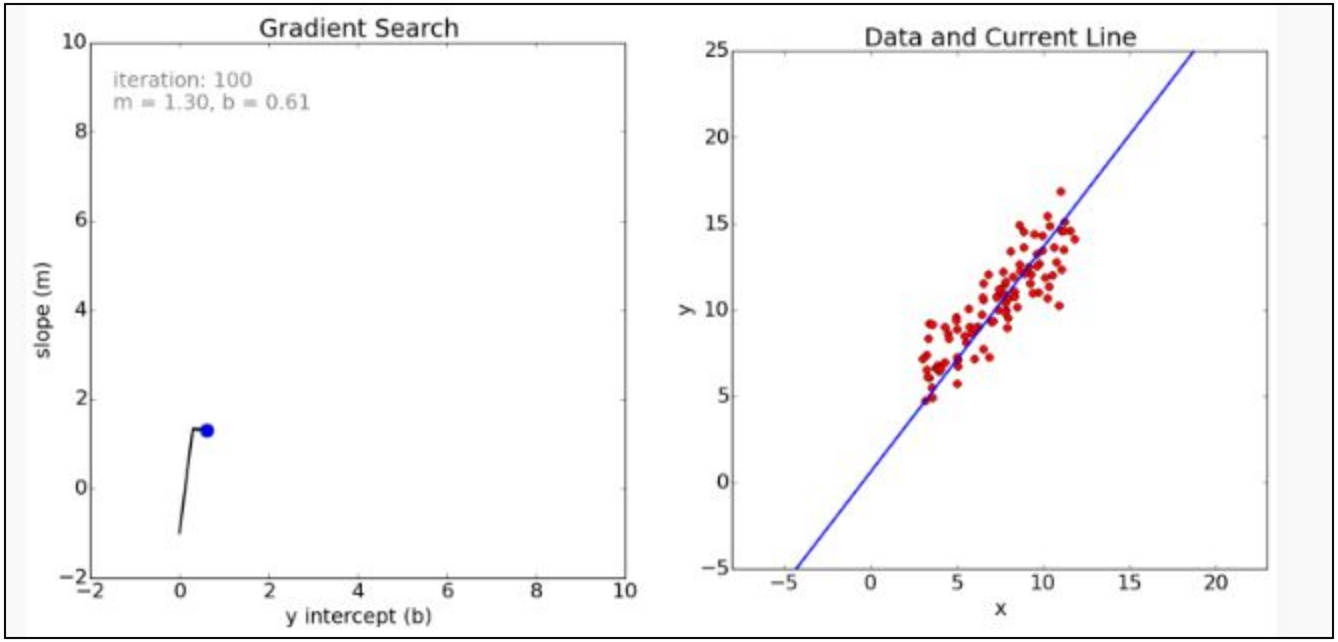
- $\Theta_0 = \begin{pmatrix} \theta_0 = 0 \\ \theta_1 = 1 \end{pmatrix}, \eta = 0.01$

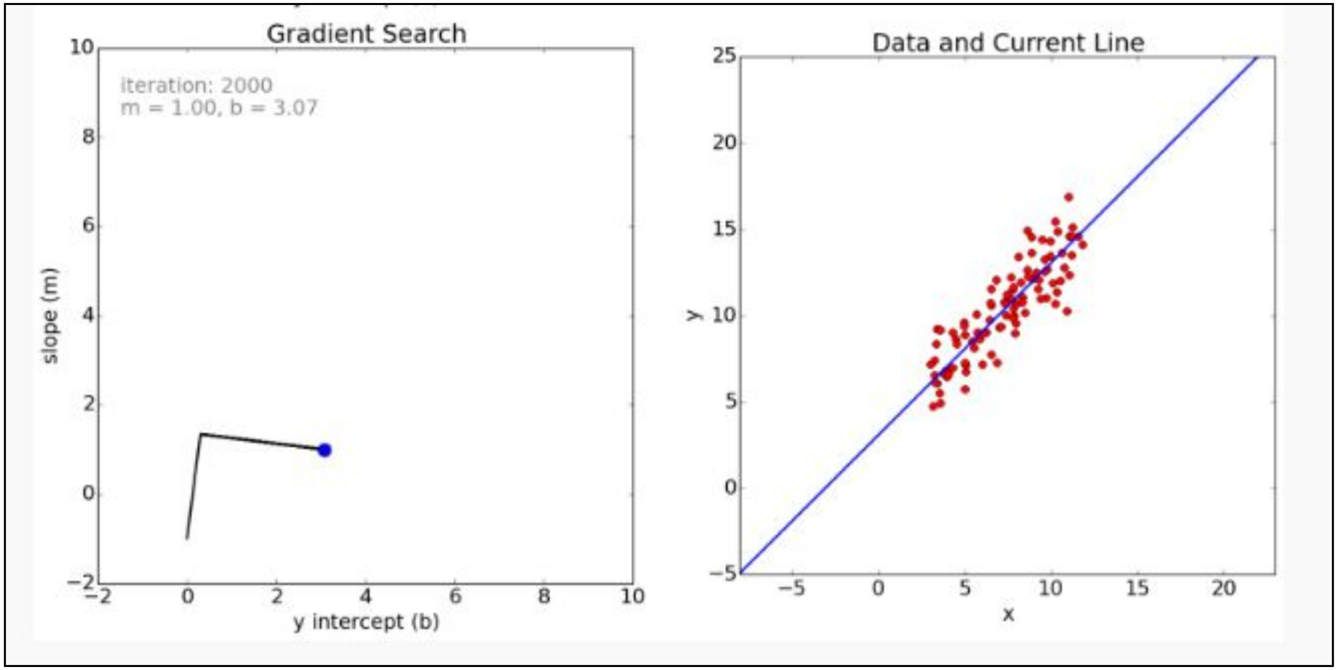


In 1D problem, often we call, $\theta_0 = \text{y-intercept} = b$
$\theta_1 = \text{slope} = m.$
Line equation is $y = mx + b$


Gradient Search
iteration: 1
m = 0.40, b = 0.18
slope (m)
y intercept (b)
Data and Current Line
y
x


Gradient Search
iteration: 2
m = 0.96, b = 0.25
slope (m)
y intercept (b)
Data and Current Line
y
x


Gradient Search
iteration: 100
m = 1.30, b = 0.61
slope (m)
y intercept (b)
Data and Current Line
y
x


Gradient Search
iteration: 2000
m = 1.00, b = 3.07
slope (m)
y intercept (b)
Data and Current Line
y
x

When to stop

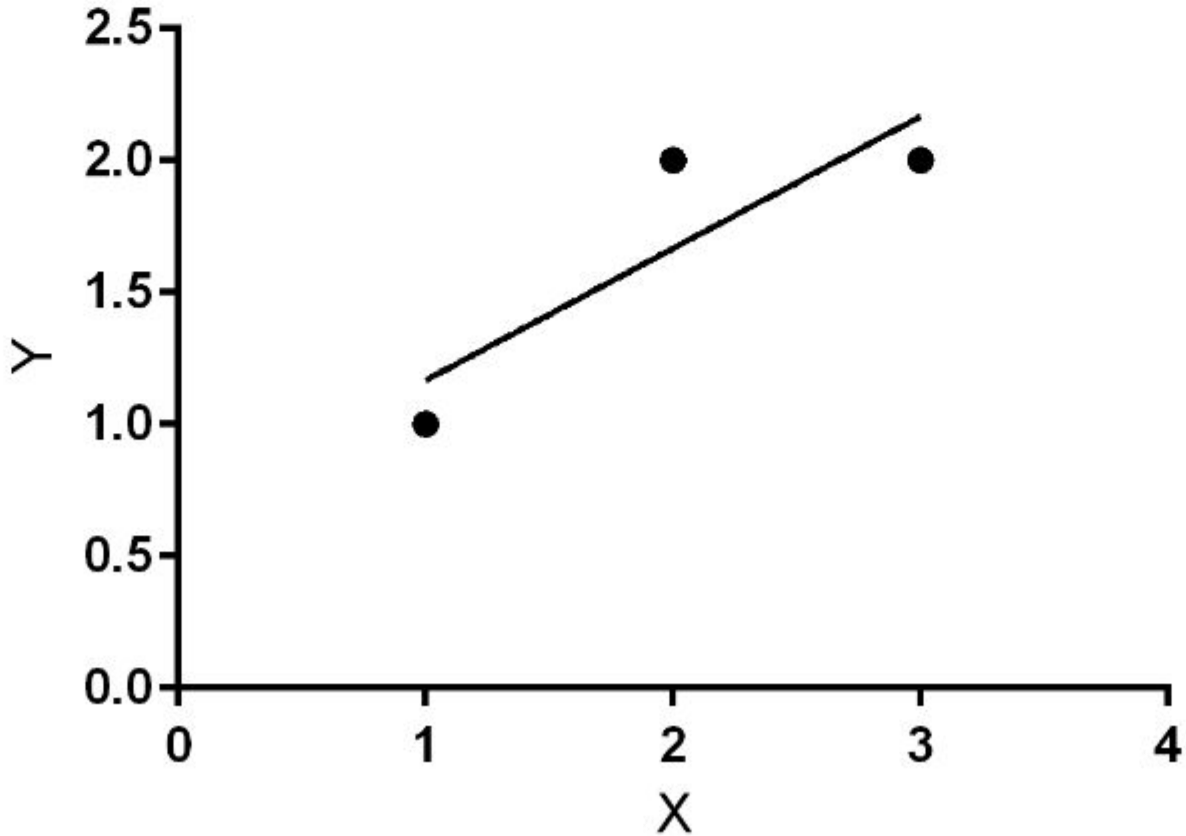
- Ideally when $\nabla J = 0$
- In practice, often when $||\nabla J|| < \epsilon$ where ϵ is a small real number like 0.01 is chosen as the stopping condition.

Example

- Given $D = \{(1,1), (2,2), (3,2)\}$
- Let us begin with $\Theta_0 = \begin{pmatrix} \theta_0 = 0 \\ \theta_1 = 1 \end{pmatrix}$,
- Let $\eta = 0.1$

- Can you do this?

- Solution is, $y = \frac{2}{3} + \frac{x}{2}$



GENERALIZING TO MULTIVARIATE DATA

Notation

- Let the given data is $\mathrm{D} = \{(X_1, y_1), \dots, (X_n, y_n)\}$.
- Let $X_i \in R^d$ and $y_i \in R$
- Further, $X_i = [x_{i1} \quad \cdots \quad x_{id}]^t$
- We augment each X_i with 1 in order to simplify.
- The augmented vector we call Z_i

-
- $Z_i = [1 \ \ x_{i1} \ \cdots \ x_{id}]^t = \begin{bmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{id} \end{bmatrix}$

We like to solve

- $$\begin{gathered}\theta_0 + \theta_1 x_{11} + \cdots + \theta_d x_{1d} = y_1 \\ \theta_0 + \theta_1 x_{21} + \cdots + \theta_d x_{2d} = y_2 \\ \vdots \\ \theta_0 + \theta_1 x_{n1} + \cdots + \theta_d x_{nd} = y_n\end{gathered}$$

In matrix form $\boldsymbol{Z}\Theta = Y$

-

- $Z_i = \begin{bmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{id} \end{bmatrix} = [1 \quad x_{i1} \quad \dots \quad x_{id}]^t$

- $\boldsymbol{Z} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{bmatrix} = \begin{bmatrix} -Z_1 - \\ \vdots \\ -Z_n - \end{bmatrix} = \begin{bmatrix} Z_1^t \\ \vdots \\ Z_n^t \end{bmatrix}$

$$Z\Theta = Y$$

-
- $\boldsymbol{Z} = \begin{bmatrix} -Z_1 - \\ \vdots \\ -Z_n - \end{bmatrix} = \begin{bmatrix} Z_1^t \\ \vdots \\ Z_n^t \end{bmatrix}$

- $Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

- $\Theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$

Criterion J

- $J(\Theta) = \|\mathbf{Z}\Theta - Y\|^2 = (\mathbf{Z}\Theta - Y)^t(\mathbf{Z}\Theta - Y)$

$$J(\Theta) = \Theta^t\mathbf{Z}^t\mathbf{Z}\Theta - 2(\mathbf{Z}\Theta)^tY - Y^tY$$

$$J(\Theta) = \Theta^t \mathbf{Z}^t \mathbf{Z}\Theta - 2(\mathbf{Z}\Theta)^t Y - Y^t Y$$

- $$\nabla_\Theta(J) = 2(\mathbf{Z}^t \mathbf{Z})\Theta - 2\mathbf{Z}^{\mathrm{t}} Y$$

Recall by equating the above to 0, we got the closed form solution which is,

$$\Theta = (\mathbf{Z}^t \mathbf{Z})^{-1} \mathbf{Z}^t Y$$

Iterative solution

- $\Theta_{k+1} = \Theta_k - \eta\nabla(J)$
- Note, this Gradient is at Θ_k

Batch Learning

- In each iteration entire training set is considered.
- $\Theta_{k+1} = \Theta_k - \eta(2(\mathbf{Z}^t\mathbf{Z})\Theta_k - 2\mathbf{Z}^t Y)$
  $= \Theta_k - 2\eta\mathbf{Z}^t(\mathbf{Z}\Theta_k - Y)$

- Each element of Θ can be updated as below.
- $\theta_j = \theta_j - 2\eta\sum_{i=1}^{n}(h(Z_i) - y_i)x_j$

Stochastic Gradient Descent

- This is single sample correction method
- Consider one example, viz., (X_i, y_i)
- For X_i, its augmented vector is Z_i
- $Z_i = \begin{bmatrix} 1 \\ X_i \end{bmatrix}$
- $\Theta_{k+1} = \Theta_k - \eta\left(2\left(Z_i Z_i^t\right)\Theta_k - 2y_i Z_i\right)$

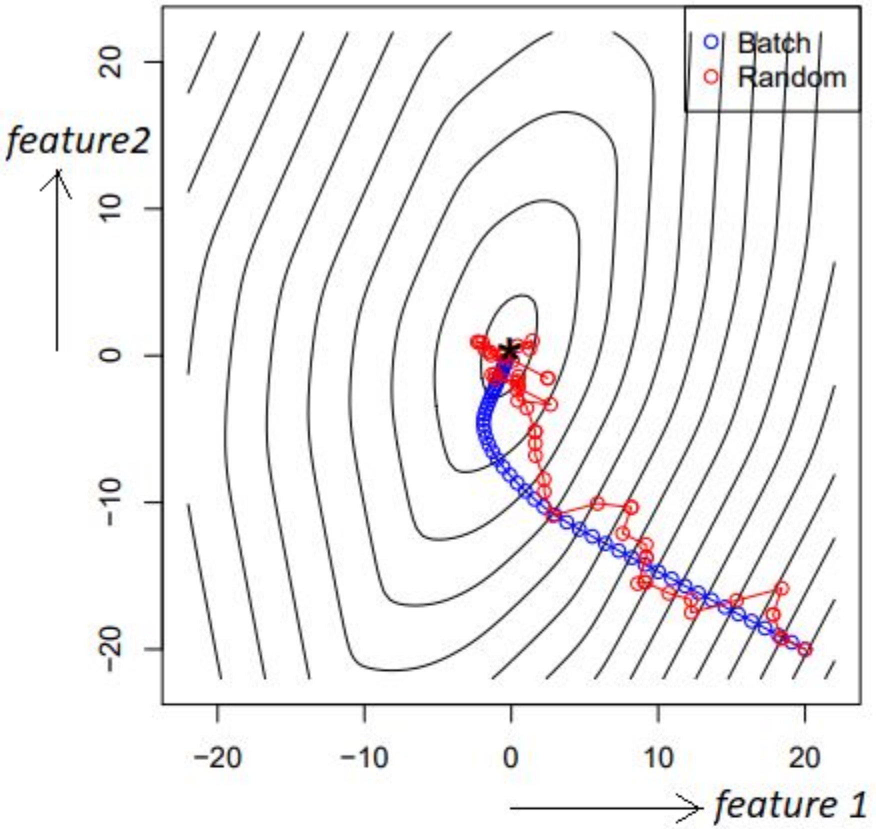
Normally X_i is chosen randomly from the training set.

Stochastic Gradient Descent

- $\Theta_{k+1} = \Theta_k - \eta\left(2\left(Z_i Z_i^t\right)\Theta_k - 2y_i Z_i\right)$
  $= \Theta_k - 2\eta\left(Z_i^t \Theta_k - \mathrm{y}_i\right) Z_i$

- Each element of Θ can be updated as below.
- $\theta_j = \theta_j - 2\eta(h(Z_i) - y_i)x_j$

Batch Vs Stochastic Convergence



Blue: batch steps,
Red: stochastic steps,

Stochastic is also known as random / single sample correction.

- Batch method smoothly converges
- Stochastic method strays away often from the optimal path.

Stopping Condition

- It is better to have a validation set (VS) and
  - keep track of error on the VS,
  - stop when the error on the VS is not decreasing considerably
- In practice we fix number of iterations.