Gradient (descent) methods and linear regression

DSA5103 Lecture 2

Yangjing Zhang

19-Jan-2023

NUS

Today's content

1. Gradient (descent) methods
2. Linear regression with one variable
3. Linear regression with multiple variables
4. Gradient descent for linear regression
5. Normal equation for linear regression

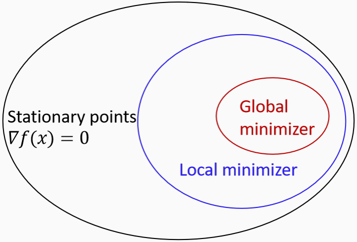
Gradient methods

Unconstrained problem

To minimize a **differentiable** function f

$$\min_{x \in \mathbb{R}^n} \quad f(x)$$

Recall that a global minimizer is a local minimizer, and a local minimizer is a stationary point



- We may try to find stationary points x, i.e., $\nabla f(x) = 0$ for solving an unconstrained problem
- When it is difficult to solve $\nabla f(x) = 0$, we look for an approximate solution via iterative methods

Algorithmic framework

A general algorithmic framework: choose $x^{(0)}$ and repeat

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad k = 0, 1, 2, \ldots$$

until some stopping criteria is satisfied

- $x^{(0)}$ initial guess of the solution
- $\alpha_k > 0$ is called the step length/step size/learning rate
- $p^{(k)}$ is a search direction

Algorithmic framework

A general algorithmic framework: choose $x^{(0)}$ and repeat

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \quad k = 0, 1, 2, \ldots$$

until some stopping criteria is satisfied

- $x^{(0)}$ initial guess of the solution
- $\alpha_k > 0$ is called the step length/step size/learning rate
- $p^{(k)}$ is a search direction
  (we hope the search direction can "improve" the iterative point in some sense)

Descent direction

The search direction $p^{(k)}$ should be a descent direction at $x^{(k)}$

- We say $p^{(k)}$ is a descent direction at $x^{(k)}$ if

$$\nabla f(x^{(k)})^T p^{(k)} < 0$$

Descent direction

The search direction $p^{(k)}$ should be a descent direction at $x^{(k)}$

- We say $p^{(k)}$ is a descent direction at $x^{(k)}$ if

$$\nabla f(x^{(k)})^T p^{(k)} < 0$$

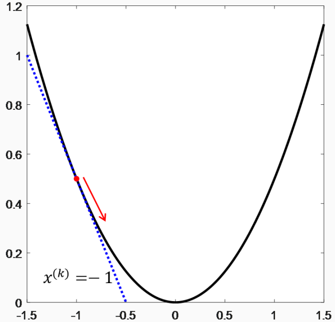
- The function value f can be reduced along this descent direction with "appropriate" step length

$$\exists\, \delta > 0 \text{ such that } \quad f(x^{(k)} + \alpha_k p^{(k)}) < f(x^{(k)}) \quad \forall\, \alpha_k \in (0, \delta)$$

Example

Example 1:

$\min_{x \in \mathbb{R}} f(x) = \frac{1}{2}x^2$



At $x^{(k)} = -1$, $p^{(k)} = 1$ is a descent direction since

$$\nabla f(x^{(k)}) = x^{(k)} = -1, \quad \nabla f(x^{(k)})^T p^{(k)} = (-1) \times 1 = -1 < 0$$

We observe that for $\alpha_k \in (0, 2)$

$$f(x^{(k)} + \alpha_k p^{(k)}) < f(x^{(k)})$$

Example

Example 2: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2.$

At $x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, show that $p^{(0)} = -\nabla f(x^{(0)})$ is a descent direction.

Example

Example 2: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2.$

At $x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, show that $p^{(0)} = -\nabla f(x^{(0)})$ is a descent direction.

Solution. Compute the gradient $\nabla f(x) = \begin{bmatrix} 2x_1 - 2x_2 \\ 4x_2 - 2x_1 - 2 \end{bmatrix}$.

Then $\nabla f(x^{(0)}) = \nabla f(0;0) = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$ and $p^{(0)} = -\nabla f(x^{(0)}) = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$.

Since $\nabla f(x^{(0)})^T p^{(0)} = [0 - 2] \begin{bmatrix} 0 \\ 2 \end{bmatrix} = -4 < 0$, $p^{(0)}$ is a descent direction at $x^{(0)}$.

Example

Example 2: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2.$

At $x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, show that $p^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $p^{(0)} = \begin{bmatrix} -2 \\ 0.1 \end{bmatrix}$ are descent directions.

Example

Example 2: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$.

At $x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, show that $p^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $p^{(0)} = \begin{bmatrix} -2 \\ 0.1 \end{bmatrix}$ are descent directions.

Solution. $\nabla f(x^{(0)}) = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$

$$\nabla f(x^{(0)})^T p^{(0)} = [0 - 2] \begin{bmatrix} 1 \\ 1 \end{bmatrix} = -2 < 0$$

$$\nabla f(x^{(0)})^T p^{(0)} = [0 - 2] \begin{bmatrix} -2 \\ 0.1 \end{bmatrix} = -0.2 < 0$$

Example

Example 2: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$.

At $x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, show that $p^{(0)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$ is not a descent direction.

Example

Example 2: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} \ f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$.

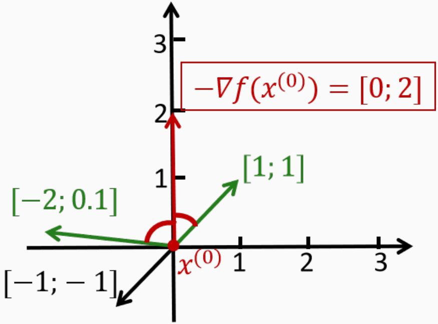
At $x^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, show that $p^{(0)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$ is not a descent direction.

Solution. $\nabla f(x^{(0)}) = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$

Since $\nabla f(x^{(0)})^T p^{(0)} = [0 - 2] \begin{bmatrix} -1 \\ -1 \end{bmatrix} = 2 > 0$, $p^{(0)}$ is not a descent direction at $x^{(0)}$.

[Exercise] Construct another descent direction.

Example



Example 2: at $x^{(0)} = [0; 0]$

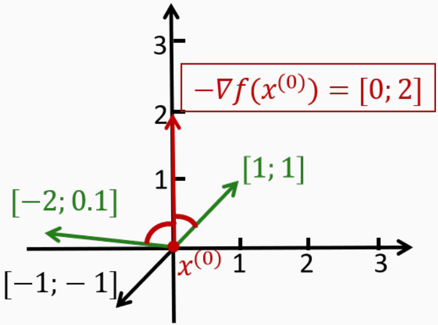
Descent directions:

$[0; 2], [1; 1], [-2; 0.1]$

Not descent directions:

$[-1; -1]$

Example



Example 2: at $x^{(0)} = [0; 0]$
Descent directions:
$[0; 2], [1; 1], [-2; 0.1]$
Not descent directions:
$[-1; -1]$

- There can be infinitely many descent directions
- The "angle" between a descent direction and $-\nabla f(\cdot)$ is less than 90^o

- Among all directions, the value of f decreases most rapidly along the direction $-\nabla f(\cdot)$
- The direction $-\nabla f(\cdot)$ is known as the steepest descent direction

Steepest descent method

Algorithm (Steepest descent method)

Choose $x^{(0)}$ and $\epsilon > 0$; Set $k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \epsilon$ **do**

find the step length α_k (e.g., by certain line search rule)

$x^{(k+1)} = x^{(k)} - \alpha_k \nabla f(x^{(k)})$

$k \leftarrow k + 1$

end(while)

return $x^{(k)}$

One may choose to use a constant step length (say $\alpha_k = 0.1$), or find it via line search rules:

- Exact line search
- Backtracking line search

Constant step length

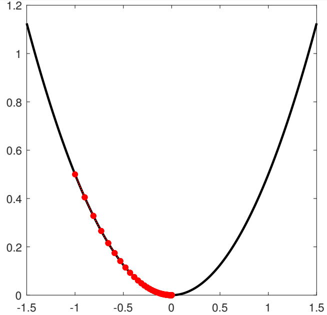
Example 3: $\min_{x \in \mathbb{R}} f(x) = \frac{1}{2}x^2$

Apply steepest descent method with $x^{(0)} = -1$, $\epsilon = 10^{-4}$, and constant step length $\alpha_k = 0.1$

Constant step length

Example 3: $\min\limits_{x \in \mathbb{R}} f(x) = \frac{1}{2}x^2$

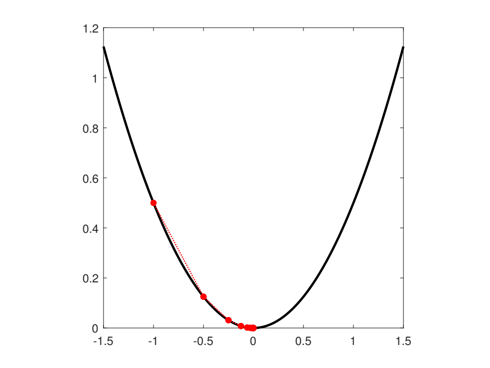
Apply steepest descent method with $x^{(0)} = -1$, $\epsilon = 10^{-4}$, and constant step length $\alpha_k = 0.1$



- Return $x^{(k)} = -9.40 \times 10^{-5}$ at iteration $k = 89$
- When the step length is too small, the method can be slow
- As it approaches the minimizer, the method will automatically take smaller steps

Constant step length

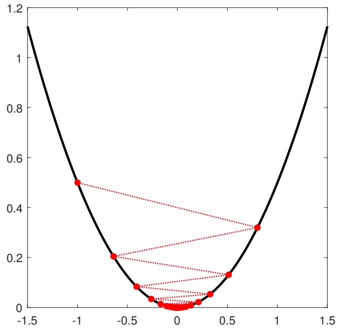
Example 3: constant step length $\alpha_k = 0.5$



- Return $x^{(k)} = -6.10 \times 10^{-5}$ at iteration $k = 15$
- In this particular example, $\alpha_k = 0.5$ (converge in 15 steps) is better than $\alpha_k = 0.1$ (converge in 89 steps)

Constant step length

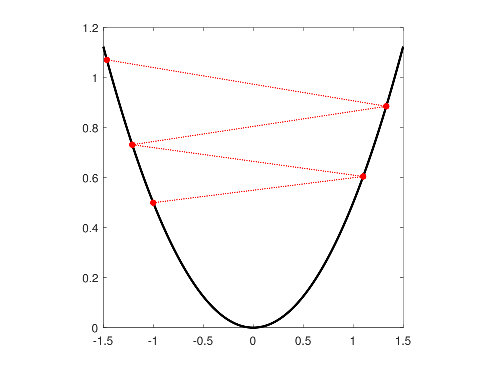
Example 3: constant step length $\alpha_k = 1.8$



- Return $x^{(k)} = -8.51 \times 10^{-5}$ at iteration $k = 43$
- Due to the big step length, the iterates are oscillating around the solution but still converge

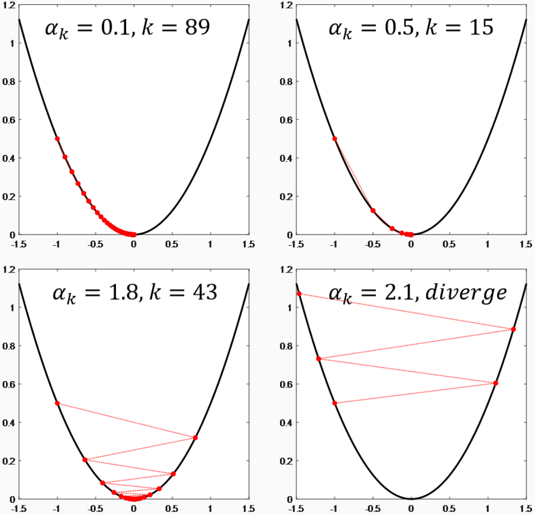
Constant step length

Example 3: constant step length $\alpha_k = 2.1$



- The step length is too large, and the method diverges

Constant step length



Exact line search

Exact line search tries to find α_k by solving the one-dimensional problem

$$\min_{\alpha > 0} \quad \phi(\alpha) := f(x^{(k)} + \alpha p^{(k)})$$

- In general, exact line search is the most difficult part of the steepest descent method
- If f is a simple function, it may be possible to obtain an analytical solution for α_k by solving $\phi'(\alpha) = 0$

Example

Example 4: $\min_{x \in \mathbb{R}} f(x) = \frac{1}{2}x^2$

Apply steepest descent method with exact line search, given $x^{(0)} = -1$.

Example

Example 4: $\min_{x \in \mathbb{R}} f(x) = \frac{1}{2}x^2$

Apply steepest descent method with exact line search, given $x^{(0)} = -1$.

Solution. $k = 0$, $p^{(0)} = -\nabla f(x^{(0)}) = -\nabla f(-1) = 1$. Find α_0 by solving

$$\min_{\alpha > 0} \quad \phi(\alpha) = f(x^{(0)} + \alpha p^{(0)}) = \frac{1}{2}(\alpha - 1)^2.$$

Obviously, $\alpha_0 = 1$, and $x^{(1)} = x^{(0)} + \alpha_0 p^{(0)} = 0$ is actually the global minimizer.

Example

Example 5: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + x_2^2 + 2x_1 + 4$ (A convex problem)

Apply steepest descent method with exact line search, given $x^{(0)} = [2;1]$.

Example

Example 5: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + x_2^2 + 2x_1 + 4$ (A convex problem)

Apply steepest descent method with exact line search, given $x^{(0)} = [2;1]$.

Solution: Calculate the gradient $\nabla f(x) = \begin{bmatrix} 2x_1 + 2 \\ 2x_2 \end{bmatrix}, \nabla f(x^{(0)}) = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$.

Then $p^{(0)} = -\nabla f(x^{(0)}) = [-6;-2]$, and find α_0 by solving

$$\begin{aligned} \min_{\alpha>0} \quad \phi(\alpha) &= f(x^{(0)} + \alpha p^{(0)}) = f(2-6\alpha; 1-2\alpha) \\ &= (2-6\alpha)^2 + (1-2\alpha)^2 + 2(2-6\alpha) + 4 \end{aligned}$$

ϕ is quadratic (convex). By setting $\phi'(\alpha) = 0$, we obtain $\alpha_0 = 0.5$. And $x^{(1)} = x^{(0)} + \alpha_0 p^{(0)} = [-1;0]$. The steepest descent method will terminate since $\nabla f(x^{(1)}) = [0;0]$.

[Exercise] Verify that $[-1;0]$ is a global minimizer (by sufficient condition and convexity).

Example

Example 6: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$ (convex)

Apply steepest descent method with exact line search, given $x^{(0)} = [0;0]$. Compute $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$.

Example

Example 6: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$ (convex)

Apply steepest descent method with exact line search, given $x^{(0)} = [0; 0]$. Compute $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$.

Solution. Compute the gradient $\nabla f(x) = \begin{bmatrix} 2x_1 - 2x_2 \\ 4x_2 - 2x_1 - 2 \end{bmatrix}$.

$\underline{k=0:}$ $p^{(0)} = -\nabla f(x^{(0)}) = [0; 2]$

$\min_{\alpha>0} \quad \phi(\alpha) = f(x^{(0)} + \alpha p^{(0)}) = f(0; 2\alpha) = 8\alpha^2 - 4\alpha$, ϕ is convex.

We let $0 = \phi'(\alpha) = 16\alpha - 4$, and obtain that $\alpha_0 = \frac{1}{4}$.

$x^{(1)} = x^{(0)} + \alpha_0 p^{(0)} = [0; \frac{1}{2}]$

Example

Example 6: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} \; f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$ (convex)

Apply steepest descent method with exact line search, $x^{(0)} = [0;0]$. Compute $x^{(1)}$, $x^{(2)}$, and $x^{(3)}$.

Solution. $\nabla f(x) = \begin{bmatrix} 2x_1 - 2x_2 \\ 4x_2 - 2x_1 - 2 \end{bmatrix}$, $x^{(1)} = [0; \frac{1}{2}]$.

$\underline{k=1:}$ $p^{(1)} = -\nabla f(x^{(1)}) = [1;0]$

$\min_{\alpha>0} \quad \phi(\alpha) = f(x^{(1)} + \alpha p^{(1)}) = f(\alpha; \frac{1}{2}) = \alpha^2 - \alpha - \frac{1}{2}$, ϕ is convex.

We let $0 = \phi'(\alpha) = 2\alpha - 1$, and obtain that $\alpha_1 = \frac{1}{2}$.

$x^{(2)} = x^{(1)} + \alpha_1 p^{(1)} = [\frac{1}{2}; \frac{1}{2}]$

$\underline{k=2:}$ $x^{(3)} = [\frac{1}{2}; \frac{3}{4}]$ [Exercise]

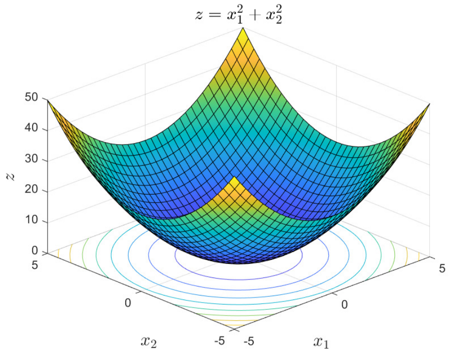
In fact, the global minimizer ($x^* = [1;1]$) can be found by solving $\nabla f(x) = 0$.

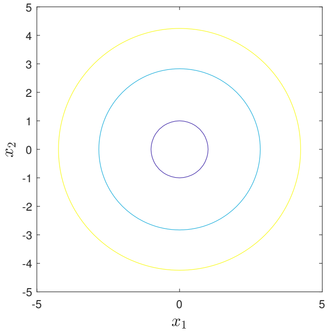
Contour plot

A contour is a fixed height $f(x_1, x_2) = c$.

Left: plot $z = f(x_1, x_2) = x_1^2 + x_2^2$

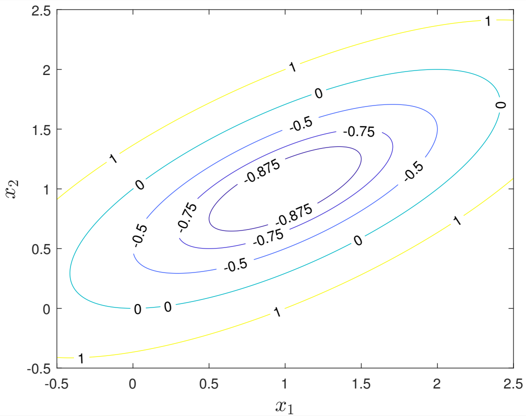
Right: contour plot $f(x_1, x_2) = 1$, $f(x_1, x_2) = 8$, $f(x_1, x_2) = 18$





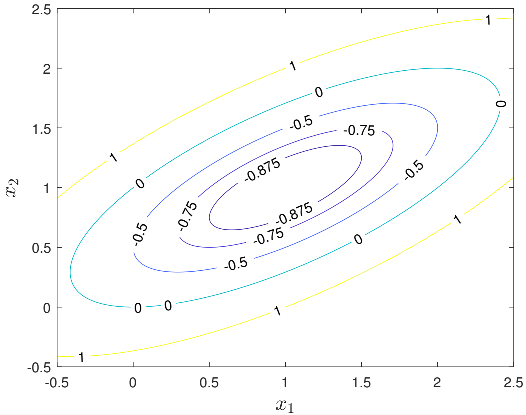
Contour plot

Contour plot of $f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$ in Example 6



Contour plot

Contour plot of $f(x) = x_1^2 + 2x_2^2 - 2x_1x_2 - 2x_2$ in Example 6



$$x^{(0)} = [0; 0]$$
$$x^{(1)} = [0; \tfrac{1}{2}]$$
$$x^{(2)} = [\tfrac{1}{2}; \tfrac{1}{2}]$$
$$x^{(3)} = [\tfrac{1}{2}; \tfrac{3}{4}]$$
$$\ldots$$

Steepest descent method with exact line search follows a zig-zag path towards the solution. This zigzagging makes the method inherently slow.

Steepest descent method with exact line search

Algorithm (Steepest descent method with exact line search)

Choose $x^{(0)}$ and $\epsilon > 0$; Set $k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \epsilon$ **do**

$p^{(k)} = -\nabla f(x^{(k)})$

$\alpha_k = \arg\min_{\alpha > 0} \quad \phi(\alpha) = f(x^{(k)} + \alpha p^{(k)})$

$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$

$k \leftarrow k + 1$

end(while)

return $x^{(k)}$

Properties of steepest descent method with exact line search*

Let $\{x^{(k)}\}$ be the sequence generated by steepest descent method with exact line search

- The steepest descent method with exact line search moves in perpendicular steps. More precisely, $x^{(k)} - x^{(k+1)}$ is orthogonal (perpendicular) to $x^{(k+1)} - x^{(k+2)}$.
- Monotonic decreasing property:

$$f(x^{(k+1)}) < f(x^{(k)}) \text{ if } \nabla f(x^{(k)}) \neq 0.$$

- Suppose f is a coercive[1] function with continuous first order derivatives on $\mathbb{R}^n$. Then some subsequence of $\{x^{(k)}\}$ converges. The limit of any convergent subsequence of $\{x^{(k)}\}$ is a stationary point of f.

[1] A continuous function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be coercive if $\lim_{\|x\| \to \infty} f(x) = +\infty$

Backtracking line search

Backtracking line search starts with a relatively large step length and iteratively shrinks it (i.e., "backtracking") until the Armijo condition holds.

Algorithm (Backtracking line search)

Choose $\bar{\alpha} > 0$, $\rho \in (0,1)$, $c_1 \in (0,1)$; Set $\alpha \leftarrow \bar{\alpha}$

repeat until $\underbrace{f(x^{(k)} + \alpha p^{(k)}) \leq f(x^{(k)}) + c_1 \alpha \nabla f(x^{(k)})^T p^{(k)}}_{\text{Armijo condition}}$

$\alpha \leftarrow \rho\alpha$

end(repeat)

return $\alpha_k = \alpha$

Backtracking line search

- Note that $p^{(k)}$ is a descent direction

$$\nabla f(x^{(k)})^T p^{(k)} < 0$$

  The Armijo condition

$$f(x^{(k)} + \alpha p^{(k)}) \leq f(x^{(k)}) + c_1 \alpha \nabla f(x^{(k)})^T p^{(k)}$$

  requires a reasonable amount of decrease in the objective function, rather than find the best step length (as in exact line search).

- For example, one can set

$$\bar{\alpha} = 1,\ \rho = 0.9,\ c_1 = 10^{-4}$$

  in practice. Namely, we start with step length 1 and continue with $0.9, 0.9^2, 0.9^3, \ldots$ until the Armijo condition is satisfied.

Example

Example 7: $\min\limits_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + x_2^2 + 2x_1 + 4$

Apply steepest descent method with backtracking line search, given $x^{(0)} = [2;1]$, $\bar{\alpha} = 1$, $\rho = 0.9$, $c_1 = 10^{-4}$. Compute $x^{(1)}$.

Example

Example 7: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} \; f(x) = x_1^2 + x_2^2 + 2x_1 + 4$

Apply steepest descent method with backtracking line search, given $x^{(0)} = [2;1]$, $\bar{\alpha} = 1$, $\rho = 0.9$, $c_1 = 10^{-4}$. Compute $x^{(1)}$.

Solution: Calculate the gradient $\nabla f(x) = \begin{bmatrix} 2x_1 + 2 \\ 2x_2 \end{bmatrix}, \nabla f(x^{(0)}) = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$.

$\underline{k = 0.}$ $p^{(0)} = -\nabla f(x^{(0)}) = [-6; -2]$. Do backtracking line search:

1. $\underline{\alpha = \bar{\alpha} = 1.}$ Check Armijo condition
   $f(x^{(0)} + \alpha p^{(0)}) \leq f(x^{(0)}) + c_1 \alpha \nabla f(x^{(0)})^T p^{(0)}$
   LHS $= f(-4; -1) = (-4)^2 + (-1)^2 + 2(-4) + 4 = 13$
   RHS $= (2^2 + 1^2 + 2 \cdot 2 + 4) + 10^{-4} \cdot 1 \cdot [6, 2] \begin{bmatrix} -6 \\ -2 \end{bmatrix} = 12.996$
   Armijo condition fails.

Example

Example 7: $\min_{x=(x_1;x_2)\in\mathbb{R}^2} f(x) = x_1^2 + x_2^2 + 2x_1 + 4$

Apply steepest descent method with backtracking line search, given $x^{(0)} = [2; 1]$, $\bar{\alpha} = 1$, $\rho = 0.9$, $c_1 = 10^{-4}$. Compute $x^{(1)}$.

Solution:

$\underline{k = 0.}$ $p^{(0)} = -\nabla f(x^{(0)}) = [-6; -2]$. Do backtracking line search:

2. $\underline{\alpha = \rho\bar{\alpha} = 0.9}$. Check Armijo condition
   $f(x^{(0)} + \alpha p^{(0)}) \leq f(x^{(0)}) + c_1 \alpha \nabla f(x^{(0)})^T p^{(0)}$
   LHS $= f(-3.4; -0.8) = (-3.4)^2 + (-0.8)^2 + 2(-3.4) + 4 = 9.4$
   RHS $= (2^2 + 1^2 + 2 \cdot 2 + 4) + 10^{-4} \cdot 0.9 \cdot [6, 2] \begin{bmatrix} -6 \\ -2 \end{bmatrix} = 12.9964$
   Armijo condition holds. Set $\alpha_0 = 0.9$.

New iterate $x^{(1)} = [-3.4; -0.8]$.

Steepest descent method with backtracking line search

Algorithm (Steepest descent method with backtracking line search)

Choose $x^{(0)}$, $\epsilon > 0$, $\bar{\alpha} > 0$, $\rho \in (0,1)$, $c_1 \in (0,1)$; Set $k \leftarrow 0$

while $\|\nabla f(x^{(k)})\| > \epsilon$ **do**

$p^{(k)} = -\nabla f(x^{(k)})$

$\alpha \leftarrow \bar{\alpha}$

repeat until $f(x^{(k)} + \alpha p^{(k)}) \leq f(x^{(k)}) + c_1 \alpha \nabla f(x^{(k)})^T p^{(k)}$

$\alpha \leftarrow \rho\alpha$

end(repeat)

$\alpha_k \leftarrow \alpha$

$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$

$k \leftarrow k + 1$

end(while)

return $x^{(k)}$

Linear regression with one variable

Housing prices data

- Predict the price for house with area 3500

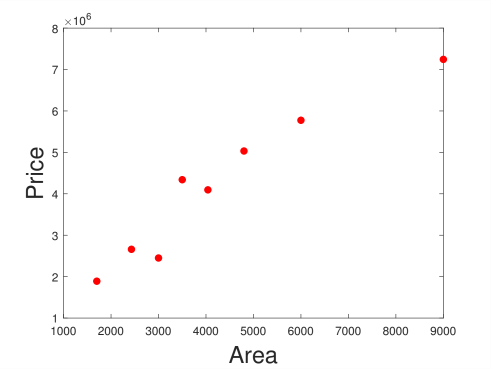


Figure 1: Housing prices data[2]

[2] https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

Housing prices data

Training set of housing prices

area (x)	**price** (y)
7420	13300000
8960	12250000
$\vdots$	$\vdots$

- Predictor/feature/"input" vector x
- Response/target/"output" variable y
- i-th training example (x_i, y_i)
- n: number of samples/training examples

Housing prices data

$$\boxed{\text{Housing prices data}}$$

$$\Downarrow$$

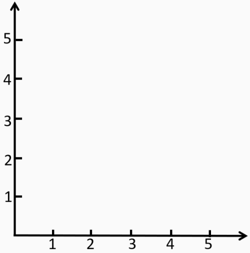
$$\boxed{\text{Learn a linear function } f(x) = f_\beta(x) = \beta_1 x + \beta_0}$$

$$\boxed{\text{input a new area of house } \hat{x}} \overset{f}{\Rightarrow} \boxed{\text{predict its price } f(\hat{x})}$$

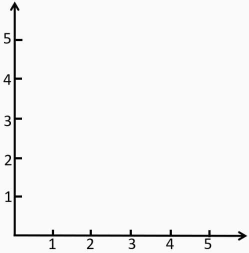
- Parameters β_1, β_0
- How to choose β_1, β_0?

Illustration

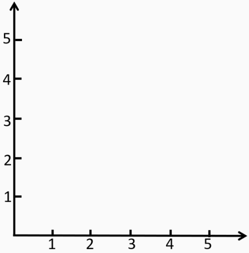
$f(x) = f_\beta(x) = \beta_1 x + \beta_0$



$\beta_1 = 1, \beta_0 = 2$



$\beta_1 = 0, \beta_0 = 3$



$\beta_1 = 2, \beta_0 = 2$

Linear regression with one variable

- Want to choose β_0, β_1 such that $f(x_i)$ is close to y_i for every training example (x_i, y_i)

Linear regression with one variable

- Want to choose β_0, β_1 such that $f(x_i)$ is close to y_i for every training example (x_i, y_i)
- Want to find β_0, β_1

$$\underset{\beta_0,\beta_1}{\text{minimize}} \quad \underbrace{\frac{1}{2}\sum_{i=1}^{n}(\beta_1 x_i + \beta_0 - y_i)^2}_{\text{squared error}}$$

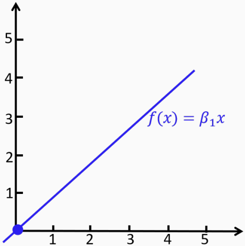
- Objective/cost/loss function

$$L(\beta_0, \beta_1) = \frac{1}{2}\sum_{i=1}^{n}(\beta_1 x_i + \beta_0 - y_i)^2$$

- Squared error function is probably the most commonly used cost function in linear regression

Understand the cost function

- Assume $\beta_0 = 0$ for simplicity
- Consider linear model $f(x) = \beta_1 x$



- Optimization model

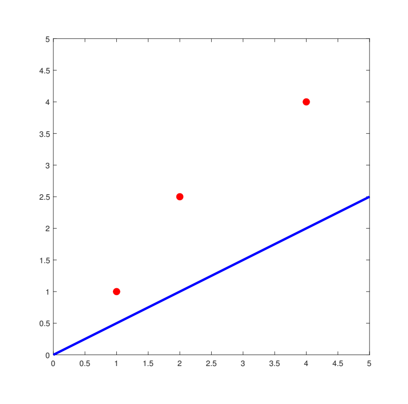
$$\underset{\beta_1}{\text{minimize}} \quad L(\beta_1) = \frac{1}{2}\sum_{i=1}^{n}(\beta_1 x_i - y_i)^2$$

Understand the cost function

Toy example

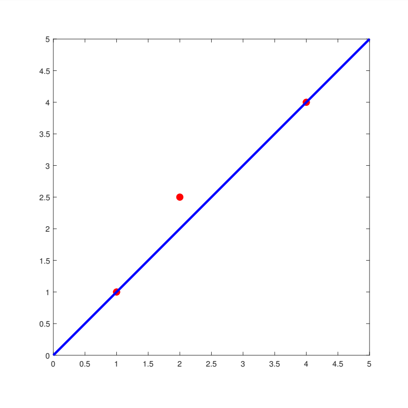
Given training data $(1,1),(2,2.5),(4,4)$ and linear model $f(x)=\beta_1 x$ (assume $\beta_0=0$). Compute the values of $L(\beta_1)=\frac{1}{2}\sum_{i=1}^{n}(\beta_1 x_i - y_i)^2$ at $\beta_1=0.5$, $\beta_1=1$, $\beta_1=1.1$.

$\beta_1 = 0.5$



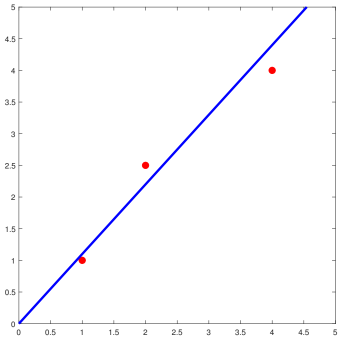
Understand the cost function

$\beta_1 = 1$

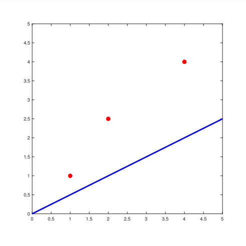


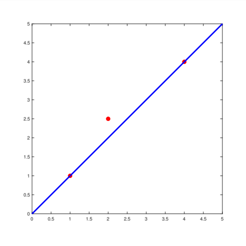
Understand the cost function

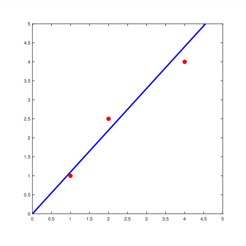
$\beta_1 = 1.1$

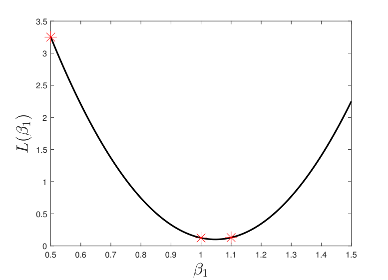


Understand the cost function









- $\beta_1 = 0.5 \quad L(\beta_1) = 3.25$
- $\beta_1 = 1 \qquad L(\beta_1) = 0.125$
- $\beta_1 = 1.1 \quad L(\beta_1) = 0.13$

Steepest descent method for univariate linear regression

- Linear regression with one variable aims to find β_0, β_1

$$\underset{\beta_0, \beta_1}{\text{minimize}} \quad L(\beta_0, \beta_1) = \frac{1}{2} \sum_{i=1}^{n} (\beta_1 x_i + \beta_0 - y_i)^2$$

- In steepest descent method, we repeat

Steepest descent method for univariate linear regression

- Linear regression with one variable aims to find β_0, β_1

$$\underset{\beta_0,\beta_1}{\text{minimize}} \quad L(\beta_0,\beta_1) = \frac{1}{2}\sum_{i=1}^{n}(\beta_1 x_i + \beta_0 - y_i)^2$$

- In steepest descent method, we repeat

$$\begin{bmatrix} \beta_0^{(k+1)} \\ \beta_1^{(k+1)} \end{bmatrix} = \begin{bmatrix} \beta_0^{(k)} \\ \beta_1^{(k)} \end{bmatrix} - \alpha_k \underbrace{\begin{bmatrix} \frac{\partial}{\partial \beta_0} L(\beta_0^{(k)}, \beta_1^{(k)}) \\ \frac{\partial}{\partial \beta_1} L(\beta_0^{(k)}, \beta_1^{(k)}) \end{bmatrix}}_{= \nabla L(\beta_0^{(k)}, \beta_1^{(k)})}$$

- Calculate

$$\frac{\partial}{\partial \beta_0} L(\beta_0,\beta_1) = \sum_{i=1}^{n}(\beta_1 x_i + \beta_0 - y_i)$$

$$\frac{\partial}{\partial \beta_1} L(\beta_0,\beta_1) = \sum_{i=1}^{n}(\beta_1 x_i + \beta_0 - y_i)x_i$$

Steepest descent method for univariate linear regression

Algorithm (Steepest descent method for univariate linear regression)

Choose $\beta_0^{(0)}$, $\beta_1^{(0)}$ and $\epsilon > 0$; Set $k \leftarrow 0$

while $\|\nabla L(\beta_0^{(k)}, \beta_1^{(k)})\| > \epsilon$ **do**

determine the step length α_k

$$\beta_0^{(k+1)} = \beta_0^{(k)} - \alpha_k \sum_{i=1}^{n} (\beta_1^{(k)} x_i + \beta_0^{(k)} - y_i)$$

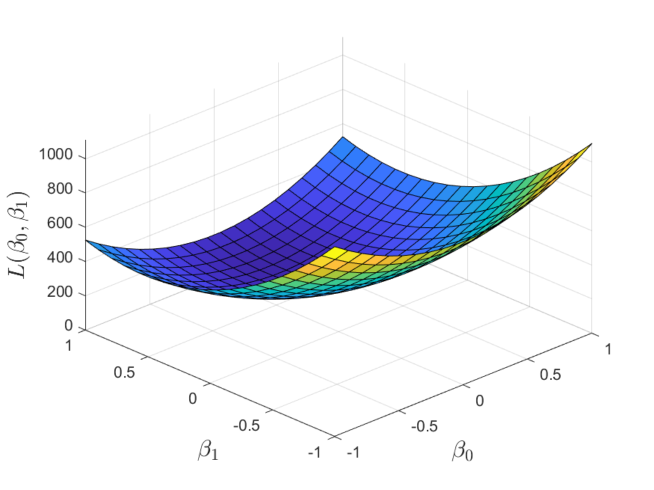
$$\beta_1^{(k+1)} = \beta_1^{(k)} - \alpha_k \sum_{i=1}^{n} (\beta_1^{(k)} x_i + \beta_0^{(k)} - y_i) x_i$$

$k \leftarrow k + 1$

end(while)

return $\beta_0^{(k)}, \beta_1^{(k)}$

Steepest descent method for univariate linear regression



Linear regression with multiple variables

Multiple features

Training set of housing prices

area	#bedrooms	#bathrooms	stories	$\cdots$	price
7420	4	2	2	$\cdots$	13300000
8960	4	4	4	$\cdots$	12250000
		$\vdots$			

- Predictor/feature/"input" vector x
- Response/target/"output" variable y
- i-th training example (x_i, y_i)
- j-th feature in i-th training example x_{ij}
- n: number of samples/training example
- p: number of features/variables

One feature vs. multiple features

One feature

Fit linear function

$$f(x) = \beta_1 x + \beta_0,\ \beta_1 \in \mathbb{R},\ \beta_0 \in \mathbb{R}$$

Cost function

$$L(\beta_0, \beta_1) = \frac{1}{2}\sum_{i=1}^{n}(\beta_1 x_i + \beta_0 - y_i)^2$$

Optimization

$$\underset{\beta_0, \beta_1}{\text{minimize}} \quad L(\beta_0, \beta_1)$$

Multiple features

Fit linear function

$$f(x) = \beta^T x + \beta_0,\ \beta \in \mathbb{R}^p,\ \beta_0 \in \mathbb{R}$$

Cost function

$$\begin{aligned} L(\beta_0, \beta) &= L(\beta_0, \beta_1, \beta_2, \ldots, \beta_p) \\ &= \frac{1}{2}\sum_{i=1}^{n}(\beta^T x_i + \beta_0 - y_i)^2 \end{aligned}$$

Optimization

$$\underset{\beta_0, \beta_1, \ldots, \beta_p}{\text{minimize}} \quad L(\beta_0, \beta_1, \ldots, \beta_p)$$

Steepest descent method for multivariate linear regression

- Linear regression with multiple variables aims to find $\beta_0, \beta_1, \ldots, \beta_p$

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \quad L(\beta_0,\beta_1,\ldots,\beta_p) = \frac{1}{2}\sum_{i=1}^{n}(\underbrace{\beta^T x_i}_{\substack{\| \\ \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}}} + \beta_0 - y_i)^2$$

Steepest descent method for multivariate linear regression

- Linear regression with multiple variables aims to find $\beta_0, \beta_1, \ldots, \beta_p$

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \quad L(\beta_0, \beta_1, \ldots, \beta_p) = \frac{1}{2}\sum_{i=1}^{n}(\underbrace{\beta^T x_i}_{\substack{\| \\ \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}}} + \beta_0 - y_i)^2$$

- Calculate

$$\begin{aligned}
\frac{\partial}{\partial \beta_0} L(\beta_0, \beta_1, \ldots, \beta_p) &= \sum_{i=1}^{n}(\beta^T x_i + \beta_0 - y_i) \\
\frac{\partial}{\partial \beta_1} L(\beta_0, \beta_1, \ldots, \beta_p) &= \sum_{i=1}^{n}(\beta^T x_i + \beta_0 - y_i)x_{i1} \\
\frac{\partial}{\partial \beta_2} L(\beta_0, \beta_1, \ldots, \beta_p) &= \sum_{i=1}^{n}(\beta^T x_i + \beta_0 - y_i)x_{i2} \\
&\vdots \\
\frac{\partial}{\partial \beta_p} L(\beta_0, \beta_1, \ldots, \beta_p) &= \sum_{i=1}^{n}(\beta^T x_i + \beta_0 - y_i)x_{ip}
\end{aligned}$$

One feature vs. multiple features

One feature

Steepest descent

$$\beta_0^{(k+1)} =$$

$$\beta_0^{(k)} - \alpha_k \sum_{i=1}^{n} (\beta_1^{(k)} x_i + \beta_0^{(k)} - y_i)$$

$$\beta_1^{(k+1)} =$$

$$\beta_1^{(k)} - \alpha_k \sum_{i=1}^{n} (\beta_1^{(k)} x_i + \beta_0^{(k)} - y_i) x_i$$

Multiple features

Steepest descent

$$\beta_0^{(k+1)} = \beta_0^{(k)} -$$

$$\alpha_k \sum_{i=1}^{n} ((\beta^{(k)})^T x_i + \beta_0^{(k)} - y_i)$$

$$\beta_j^{(k+1)} = \beta_j^{(k)} -$$

$$\alpha_k \sum_{i=1}^{n} ((\beta^{(k)})^T x_i + \beta_0^{(k)} - y_i) x_{ij}$$

for $j = 1, 2, \ldots, p$

Steepest descent method for multivariate linear regression

Algorithm (Steepest descent method for multivariate linear regression)

Choose $\beta_0^{(0)}$, $\beta^{(0)} = (\beta_1^{(0)}, \ldots, \beta_p^{(0)})^T$ and $\epsilon > 0$; Set $k \leftarrow 0$

while $\|\nabla L(\beta_0^{(k)}, \beta^{(k)})\| > \epsilon$ **do**

determine the step length α_k

$$\beta_0^{(k+1)} = \beta_0^{(k)} - \alpha_k \sum_{i=1}^{n} ((\beta^{(k)})^T x_i + \beta_0^{(k)} - y_i)$$

for $j = 1, 2, \ldots, p$

$$\beta_j^{(k+1)} = \beta_j^{(k)} - \alpha_k \sum_{i=1}^{n} ((\beta^{(k)})^T x_i + \beta_0^{(k)} - y_i) x_{ij}$$

end(for)

$k \leftarrow k + 1$

end(while)

return $\beta_0^{(k)}$, $\beta^{(k)} = (\beta_1^{(k)}, \ldots, \beta_p^{(k)})^T$

Standardization: feature scaling

area	#bedrooms	#bathrooms	stories	$\cdots$	price
7420	4	2	2	$\cdots$	13300000
8960	4	4	4	$\cdots$	12250000
		$\vdots$			

- Feature matrix: an $n \times p$ matrix X, each row is an sample, each column is a feature
- Response vector: $Y = (y_1, y_2, \ldots, y_p)^T$
- Standardization of each column of X (feature scaling) — transform all features to the same scale

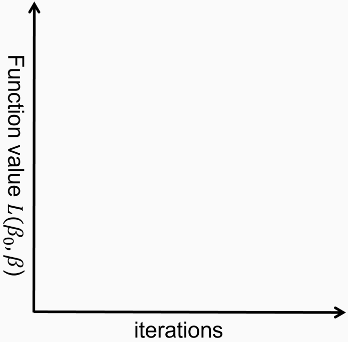
$$X_{\cdot j} = \frac{X_{\cdot j} - \text{mean}(X_{\cdot j})}{\text{standard deviation}(X_{\cdot j})}$$

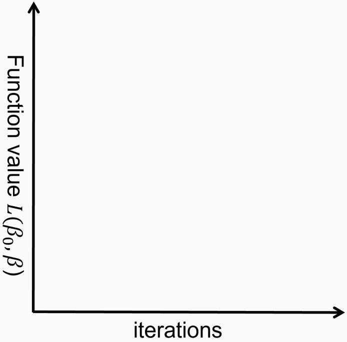
- You may also scale the response vector

$$Y = \frac{Y - \text{mean}(Y)}{\text{standard deviation}(Y)}$$

Step length

In practice, you may use backtracking line search or simply a constant step length.





Normal equation

Normal equation is to solve the linear regression problem analytically

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \quad L(\beta_0,\beta_1,\ldots,\beta_p) = \frac{1}{2}\sum_{i=1}^{n}(\beta^T x_i + \beta_0 - y_i)^2$$

- The cost function L is convex
- $\hat{\beta}$ is a global minimizer of L if and only if $\nabla L(\hat{\beta}) = 0$
- $\nabla L(\hat{\beta}) = 0$ can be written equivalently as

$$\boxed{\widehat{X}^T\widehat{X}\hat{\beta} = \widehat{X}^T Y \text{ normal equation}}$$

Notation. Recall feature matrix $X \in \mathbb{R}^{n\times p}$, response vector $Y \in \mathbb{R}^p$

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}. \text{ Define } \widehat{X} = \begin{bmatrix} 1 & x_1^T \\ 1 & x_2^T \\ 1 & \vdots \\ 1 & x_n^T \end{bmatrix} \quad \hat{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

Normal equation

Derivation*.

$$L(\hat{\beta}) = \frac{1}{2} \begin{matrix} (\beta^T x_1 + \beta_0 - y_1)^2 \\ + \\ (\beta^T x_2 + \beta_0 - y_2)^2 \\ + \\ \vdots \\ + \\ (\beta^T x_n + \beta_0 - y_n)^2 \end{matrix} = \frac{1}{2} \left\| \begin{matrix} \widehat{X}_{1\cdot}\beta - y_1 \\ \\ \widehat{X}_{2\cdot}\beta - y_2 \\ \\ \vdots \\ \\ \widehat{X}_{n\cdot}\beta - y_n \end{matrix} \right\|_F^2 = \frac{1}{2}\|\widehat{X}\hat{\beta} - Y\|_F^2$$

$$\nabla L(\hat{\beta}) = \widehat{X}^T \widehat{X}\hat{\beta} - \widehat{X}^T Y$$

Normal equation

How to solve $\boxed{\widehat{X}^T \widehat{X} \hat{\beta} = \widehat{X}^T Y \text{ normal equation}}$?

Case 1. When $\widehat{X}^T \widehat{X}$ is invertible, the normal equation implies that

$$\hat{\beta} = (\widehat{X}^T \widehat{X})^{-1} \widehat{X}^T Y$$

is the unique solution of linear regression.

This often happens when we face an over-determined system — number of training examples n is much larger than number of features p.

We have many training samples to fit but do not have enough degree of freedom.

Normal equation

How to solve $\boxed{\widehat{X}^T \widehat{X} \hat{\beta} = \widehat{X}^T Y \text{ normal equation}}$?

Case 2. When $\widehat{X}^T \widehat{X}$ is not invertible, the normal equation will have infinite number of solutions.

$\widehat{X}^T \widehat{X}$ is not invertible when we face a under-determined problem — $n < p$.

We have too many degree of freedom and do not have enough training samples.

We can apply any method for solving a linear system (e.g., Gaussian elimination) to obtain a solution.

Normal equation

Example. Give the normal equation for linear regression on the data set

feature 1	**feature 2**	**response**
1	0.2	1
0.3	4	2
5	0.6	3

Normal equation

Example. Give the normal equation for linear regression on the data set

feature 1	feature 2	response
1	0.2	1
0.3	4	2
5	0.6	3

Solution. Data $\widehat{X} = \begin{bmatrix} 1 & 1 & 0.2 \\ 1 & 0.3 & 4 \\ 1 & 5 & 0.6 \end{bmatrix}, Y = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$. Compute

$\widehat{X}^T\widehat{X} = \begin{bmatrix} 3 & 6.3 & 4.8 \\ 6.3 & 26.09 & 4.4 \\ 4.8 & 4.4 & 16.4 \end{bmatrix}, \widehat{X}^T Y = \begin{bmatrix} 6 \\ 16.6 \\ 10 \end{bmatrix}$. Then the normal equation is

$$\begin{bmatrix} 3 & 9 & 12 \\ 9 & 35 & 44 \\ 12 & 44 & 66 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 16.6 \\ 10 \end{bmatrix} \Rightarrow \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} 0.4651 \\ 0.4651 \\ 0.3488 \end{bmatrix}$$

Steepest descent vs. normal equation

Steepest descent	Normal equation
iterative method	analytical solution
need to choose step length	no need to choose step length
works well with large number of features p	solving the linear system of normal equation is slow when p is large

In practice,

$p \leq 5000$, normal equation

$p > 5000$, steepest descent