Machine Learning & Pattern Recognition

Xin Xin(辛鑫)

xinxin@sdu.edu.cn

https://xinxin-me.github.io/

Linear Regression

Linear Regression

age	23 years
annual salary	NTD 1,000,000
year in job	0.5 year
current debt	200,000

Training dataset: $\boldsymbol{\mathcal{D}} = \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \dots, (\boldsymbol{x_m}, y_m)\}$;
Features of the i-th customer: $\boldsymbol{x_i} = (x_{i1}\ x_{i2}\ \dots\ x_{id})^{\boldsymbol{T}}$; (Column vector)
The **ground truth** of the credit limit for the i-th customer: $y_i \in \mathbb{R}$ **.**

Linear regression: $h(\boldsymbol{x_i}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_i} + b = \sum_{j=1}^{d} w_j x_{ij} + b$, where $\boldsymbol{w} = (w_1\ w_2\ \dots\ w_d)^{\boldsymbol{T}} \in \mathbb{R}^{\boldsymbol{d}}$

For simplicity, the bias b can be merged into the weight $\boldsymbol{w}$:

$$h(\boldsymbol{x_i}) = \widehat{\boldsymbol{w}}^{\boldsymbol{T}}\widehat{\boldsymbol{x}}_{\boldsymbol{i}}$$

$\widehat{\boldsymbol{w}} = (b; \boldsymbol{w}) = (b\ w_1\ w_2\ \dots\ w_d) \in \mathbb{R}^{d+1}$
$\widehat{\boldsymbol{x}}_{\boldsymbol{i}} = (1;\ x_{i1}; x_{i2}; \dots; x_{id}) \in \mathbb{R}^{d+1}$

Linear Regression

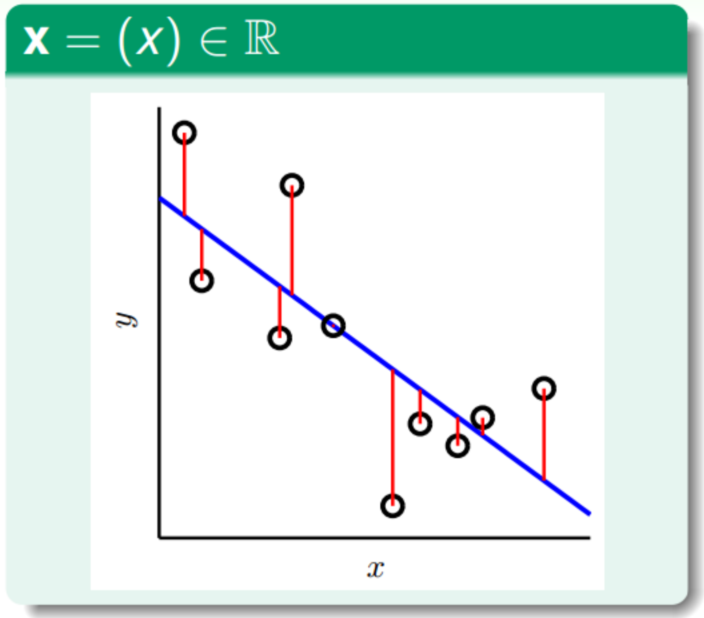
To-be-learned parameter

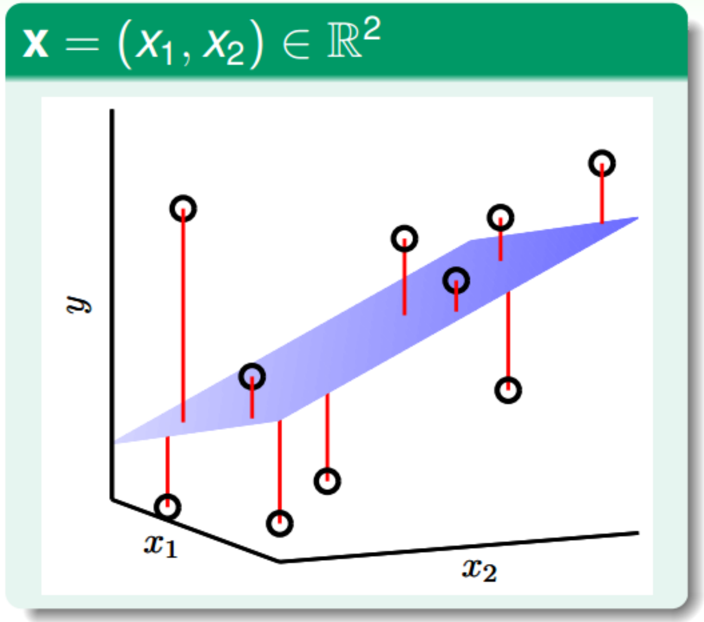
Linear regression hypothesis: $h(\boldsymbol{x_i}) = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_i} = \sum_{j=0}^{d} w_j x_{ij}\,, x_{i0} = 1$

Linear regression: find lines/hyperplanes with small residuals.

Popular/historical squared error measure:

$$L(h(\boldsymbol{x}), y) = (\hat{y} - y)^2$$





Empirical Error

We prefer to minimize the objective function where the expectation is taken across the data generating distribution p_{data} rather than just over the finite training set:

$$J^*(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim p_{data}} L(h(\boldsymbol{x}, \boldsymbol{\theta}), y)$$

However, in most cases, we do not know p_{data} but only have a training set of samples. One simplest way to convert the machine learning problem back into an optimization problem is to minimize the expected loss on the training set.

$$J(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},y)\sim \hat{P}_{data}} L(h(\boldsymbol{x}, \boldsymbol{\theta}), y)$$

Replacing the true distribution $p_{data}(\boldsymbol{x}, y)$ with the empirical distribution $\hat{P}_{data}(\boldsymbol{x}, y)$ defined by the training set.

Linear Regression

Popular/historical error measure:

squared error $L(h(\boldsymbol{x}), y) = (\hat{y} - y)^2$

$$E(\boldsymbol{w}) = \sum_{i=1}^{m} (\underbrace{h(\boldsymbol{x_i})}_{\boldsymbol{w^T}\boldsymbol{x_i}} - y_i)^2$$

Next: How to minimize $E(\boldsymbol{w})$?

Matrix Form of $E(\boldsymbol{w})$

$loss = \sum_{i=1}^{m}(\boldsymbol{w}^T\boldsymbol{x}_i - y_i)^2 + \lambda\|\boldsymbol{w}\|^2$, $\boldsymbol{w} = (w_0, w_1, \dots, w_d)^T$

$$E(\boldsymbol{w}) = \sum_{i=1}^{m}(h(\boldsymbol{x}_i) - y_i)^2 = \sum_{i=1}^{m}(\boldsymbol{w}^T\boldsymbol{x}_i - y_i)^2 = \sum_{i=1}^{m}(\boldsymbol{x}_i{}^T\boldsymbol{w} - y_i)^2$$

$$= \left\| \begin{matrix} \boldsymbol{x}_1{}^T\boldsymbol{w} - y_1 \\ \boldsymbol{x}_2{}^T\boldsymbol{w} - y_2 \\ \vdots \\ \boldsymbol{x}_m{}^T\boldsymbol{w} - y_m \end{matrix} \right\|^2 = \left\| \begin{bmatrix} - - \boldsymbol{x}_1{}^T - - \\ - - \boldsymbol{x}_2{}^T - - \\ \vdots \\ - - \boldsymbol{x}_m{}^T - - \end{bmatrix} \boldsymbol{w} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \right\|^2$$

$$= \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2$$

l_2-norm $\|\mathbf{x}\|_2 = \sqrt{x_1{}^2 + x_2{}^2 + \cdots + x_d{}^2}$

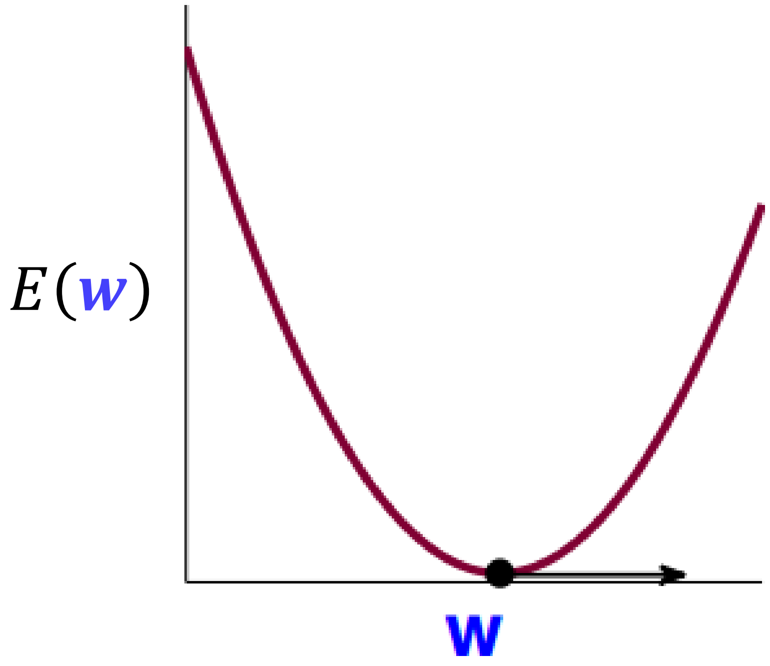
The subscript ‘2’ is usually omitted.

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{md} \end{pmatrix} \in \mathbb{R}^{m\times(d+1)}, \boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{pmatrix} \in \mathbb{R}^{d+1}, \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m$$

Matrix Form of $E(\boldsymbol{w})$

A continuous, twice differentiable function of several variables is convex on a convex set if and only if its Hessian matrix is positive semidefinite on the interior of the convex set.

$$\min E(\boldsymbol{w}) = \min ||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||^2$$



- $E(\boldsymbol{w})$: continuous, differentiable, convex
- Necessary condition of 'best' $\boldsymbol{w}$.

$$\nabla E(\boldsymbol{w}) = \begin{bmatrix} \frac{\partial E}{\partial w_0}(\boldsymbol{w}) \\ \frac{\partial E}{\partial w_1}(\boldsymbol{w}) \\ \vdots \\ \frac{\partial E}{\partial w_d}(\boldsymbol{w}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

✓ Not possible to 'roll down'

Task: find the $\boldsymbol{w}^*$ such that $\nabla E(\boldsymbol{w}^*) = 0$

The Gradient $\nabla E(w)$

$$\min_{\boldsymbol{w}} E(\boldsymbol{w}) = \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 = (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = \boldsymbol{w}^T \underbrace{\boldsymbol{X}^T\boldsymbol{X}}_{\boldsymbol{A}} \boldsymbol{w} - 2\boldsymbol{w}^T \underbrace{\boldsymbol{X}^T\boldsymbol{y}}_{\boldsymbol{b}} + \underbrace{\boldsymbol{y}^T\boldsymbol{y}}_{c}$$

One w only

$$E(w) = (aw^2 - 2bw + c)$$

$$\nabla E(w) = 2aw - 2b$$

Vector $\mathbf{w}$

$$E(\boldsymbol{w}) = (\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w} - 2\boldsymbol{w}^T\boldsymbol{b} + c)$$

$$\nabla E(w) = ?$$

Derivatives

		Differentiate		
		scalar	vector	matrix
w.r.t	scalar	scalar	vector	matrix
	vector	vector	matrix	
	matrix	matrix		

- **scalar –scalar:** e.g., $\frac{d}{dx}x^2 = 2x$
- **scalar-vector:** e.g., $f\,(\boldsymbol{x})$ is a scalar function of vector $\boldsymbol{x}$

$$\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad \frac{df}{d\mathbf{x}} = \begin{bmatrix} \frac{\sigma f}{\sigma x_1} \\ \vdots \\ \frac{\sigma f}{\sigma x_d} \end{bmatrix}$$

- **scalar-matrix:** $f\,(\boldsymbol{A})$ is a scalar function and $m \times n$ matrix $\boldsymbol{A}$

$$\frac{df}{d\mathbf{A}} = \begin{bmatrix} \frac{\sigma f}{\sigma a_{11}} & \cdots & \frac{\sigma f}{\sigma a_{1d}} \\ \vdots & \ddots & \vdots \\ \frac{\sigma f}{\sigma a_{m1}} & \cdots & \frac{\sigma f}{\sigma a_{mn}} \end{bmatrix}$$

https://en.wikipedia.org/wiki/Matrix_calculus

Matrix Calculus

- Numerator layout: lay out according to $\boldsymbol{y}$ and $\boldsymbol{x}^T$. (Jacobian formulation)
- Denominator layout: lay out according to $\boldsymbol{y}^T$ and $\boldsymbol{x}$. (Hessian formulation)

Numerator layout:
分子布局

$$\frac{\partial y}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} & \cdots & \frac{\partial y}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \frac{\partial y_2}{\partial x} \\ \vdots \\ \frac{\partial y_n}{\partial x} \end{bmatrix}$$

Denominator layout:
分母布局

$$\frac{\partial y}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix}$$

$$\frac{\partial \boldsymbol{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} & \frac{\partial y_2}{\partial x} & \cdots & \frac{\partial y_n}{\partial x} \end{bmatrix}$$

Commonly Used Derivatives

- $\frac{d}{dx}(\boldsymbol{Ax}) = \boldsymbol{A}^T$
- $\frac{d\boldsymbol{x}}{d\boldsymbol{x}} = \boldsymbol{I}$
- $\frac{d\boldsymbol{y}^T\boldsymbol{x}}{d\boldsymbol{x}} = \frac{d\boldsymbol{x}^T\boldsymbol{y}}{d\boldsymbol{x}} = \boldsymbol{y}$
- $\frac{d}{dx}(\boldsymbol{x}^T\boldsymbol{Ax}) = \begin{cases} (\boldsymbol{A} + \boldsymbol{A}^T)\boldsymbol{x} & \text{If } \mathbf{A} \text{ square} \\ 2\boldsymbol{Ax} & \text{If } \mathbf{A} \text{ symmetric} \end{cases}$

https://en.wikipedia.org/wiki/Matrix_calculus

The Gradient $\nabla E(w)$

$$\min_{\boldsymbol{w}} E(\boldsymbol{w}) = \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 = (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}) = \boldsymbol{w}^T \underbrace{\boldsymbol{X}^T\boldsymbol{X}}_{\boldsymbol{A}} \boldsymbol{w} - 2\boldsymbol{w}^T \underbrace{\boldsymbol{X}^T\boldsymbol{y}}_{\boldsymbol{b}} + \underbrace{\boldsymbol{y}^T\boldsymbol{y}}_{c}$$

One w only

$$E(w) = (aw^2 - 2bw + c)$$

$$\nabla E(w) = 2aw - 2b$$

Vector w

$$E(\boldsymbol{w}) = (\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w} - 2\boldsymbol{w}^T\boldsymbol{b} + c)$$

$$\nabla E(w) = 2\boldsymbol{A}\boldsymbol{w} - 2\boldsymbol{b}$$

$$\nabla E(w) = 2(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y})$$

Optimal Linear Regression Weights

Task: find $\boldsymbol{w}^*$ such that $\nabla E(\boldsymbol{w}^*) = 2(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y}) = \boldsymbol{0}$

Invertible/positive definite $\boldsymbol{X}^T\boldsymbol{X}$

- Unique solution

$$w^* = \underbrace{(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T}_{\text{pseudo-inverse } \boldsymbol{X}^\dagger}\boldsymbol{y}$$

Note: $\boldsymbol{X}^\dagger\boldsymbol{X} = \boldsymbol{I}$, but $\boldsymbol{X}\boldsymbol{X}^\dagger \neq \boldsymbol{I}$

If $\boldsymbol{X}$ is square and invertible, $\boldsymbol{X}^\dagger = \boldsymbol{X}^{-1}$.

Singular $\boldsymbol{X}^T\boldsymbol{X}$

- Define $\boldsymbol{X}^\dagger$ in other ways (e.g., SVD).
- Add regularization
  - E.g., l_2 norm

$\lambda > 0$

$$\min E(\boldsymbol{w}) = \min||\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}||^2 + \lambda||\boldsymbol{w}||^2$$

$$\nabla E(\boldsymbol{w}^*) = 2(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} + \lambda\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y}) = \boldsymbol{0}$$

$$\underbrace{(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})}_{\text{Invertible?}}\boldsymbol{w} = \boldsymbol{X}^T\boldsymbol{y}$$

Linear Regression Algorithm

1. From $\mathcal{D}$, construct input matrix $\boldsymbol{X}$ and output vector $\boldsymbol{y}$ by

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{md} \end{pmatrix} \in \mathbb{R}^{m\times(d+1)}, \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m$$

2. Calculate pseudo-inverse

$$\boldsymbol{X}^{\dagger} \in \mathbb{R}^{(d+1)\times m}$$

3. **Return** $\boldsymbol{w}^* = \boldsymbol{X}^{\dagger}\boldsymbol{y} \in \mathbb{R}^{(d+1)}$

Simple and efficient（？） with **good $\boldsymbol{X}^{\dagger}$**

Logistic Regression

Heart Attack Prediction Problem

age	40 years
gender	male
blood pressure	130/85
cholesterol level	240
weight	70

heart disease? **yes**

Binary classification:
Ideal $f(\boldsymbol{x}) = sign(p(+1|\boldsymbol{x}) - 0.5) \in \{-1, +1\}$

Heart Attack Prediction Problem

age	40 years
gender	male
blood pressure	130/85
cholesterol level	240
weight	70

heart attack? **80% risk**

'Soft' Binary classification:

$$f(x) = p(+1|x) \in [0,1]$$

Soft Binary classification:

Target function $f(x) = p(+1|x) \in [0,1]$

Ideal data

$$\begin{pmatrix} \mathbf{x}_1, y'_1 & = 0.9 & = P(+1|\mathbf{x}_1) \end{pmatrix}$$
$$\begin{pmatrix} \mathbf{x}_2, y'_2 & = 0.2 & = P(+1|\mathbf{x}_2) \end{pmatrix}$$
$$\vdots$$
$$\begin{pmatrix} \mathbf{x}_N, y'_N & = 0.6 & = P(+1|\mathbf{x}_N) \end{pmatrix}$$

Actual data

$$\begin{pmatrix} \mathbf{x}_1, y_1 & = \circ & \sim P(y|\mathbf{x}_1) \end{pmatrix}$$
$$\begin{pmatrix} \mathbf{x}_2, y_2 & = \times & \sim P(y|\mathbf{x}_2) \end{pmatrix}$$
$$\vdots$$
$$\begin{pmatrix} \mathbf{x}_N, y_N & = \times & \sim P(y|\mathbf{x}_N) \end{pmatrix}$$

Same data as hard binary classification, different **target function**

Logistic Hypothesis

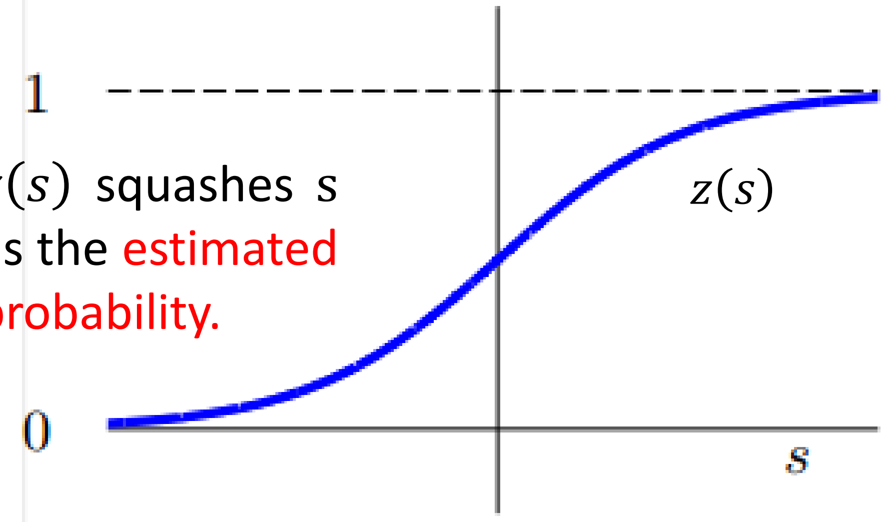
age	40 years
gender	male
blood pressure	130/85
cholesterol level	240

Let $\boldsymbol{x_i} = (x_{i0}, x_{i1}, x_{i2}, \ldots, x_{id})$ be the features of the patient, calculate a weighted 'risk score':

$$s = \sum_{j=0}^{d} w_j x_{ij} = \boldsymbol{w}^T \boldsymbol{x_i},$$

Convert the score to estimated probability by logistic function $z(s)$.

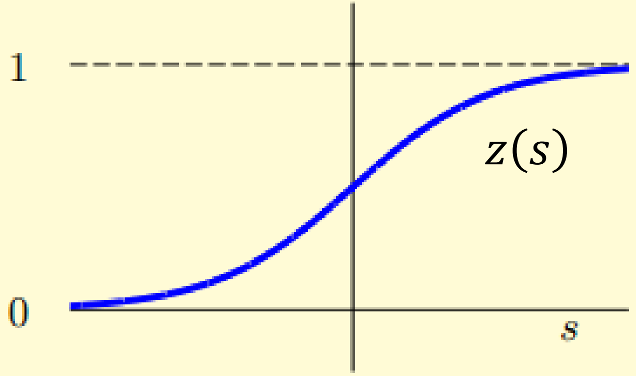
$z(s)$ squashes s as the estimated probability.



Logistic hypothesis: $h(\boldsymbol{x_i}) = z(\boldsymbol{w}^T \boldsymbol{x_i})$

Logistic Function

$$z(s) = \frac{e^{s}}{1+e^{s}} = \frac{1}{1+e^{-s}}$$



smooth, monotonic, sigmoid function of s

Bound	$z(s) \in [0,1]$ $\quad z(-\infty) = 0 \quad z(0) = 0.5 \quad z(\infty) = 1$
Symmetric	$1 - z(s) = z(-s)$
Gradient	$z'(s) = z(s)(1 - z(s))$

Logistic regression use $h(\boldsymbol{x}) = z(\boldsymbol{w}^T\boldsymbol{x})$ to approximate the target $f(\boldsymbol{x}) = p(+1|\boldsymbol{x})$

Exercise

Logistic Regression and Binary Classification

Consider any logistic hypothesis $h(\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}$ that approximates $P(y|\mathbf{x})$. 'Convert' $h(\mathbf{x})$ to a binary classification prediction by taking $\text{sign}\left(h(\mathbf{x}) - \frac{1}{2}\right)$. What is the equivalent formula for the binary classification prediction?

1. $\text{sign}\left(\mathbf{w}^T\mathbf{x} - \frac{1}{2}\right)$
2. $\text{sign}\left(\mathbf{w}^T\mathbf{x}\right)$
3. $\text{sign}\left(\mathbf{w}^T\mathbf{x} + \frac{1}{2}\right)$
4. none of the above

Exercise

Logistic Regression and Binary Classification

Consider any logistic hypothesis $h(\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}$ that approximates $P(y|\mathbf{x})$. 'Convert' $h(\mathbf{x})$ to a binary classification prediction by taking $\text{sign}\left(h(\mathbf{x}) - \frac{1}{2}\right)$. What is the equivalent formula for the binary classification prediction?

1. $\text{sign}\left(\mathbf{w}^T\mathbf{x} - \frac{1}{2}\right)$
2. $\text{sign}\left(\mathbf{w}^T\mathbf{x}\right)$
3. $\text{sign}\left(\mathbf{w}^T\mathbf{x} + \frac{1}{2}\right)$
4. none of the above

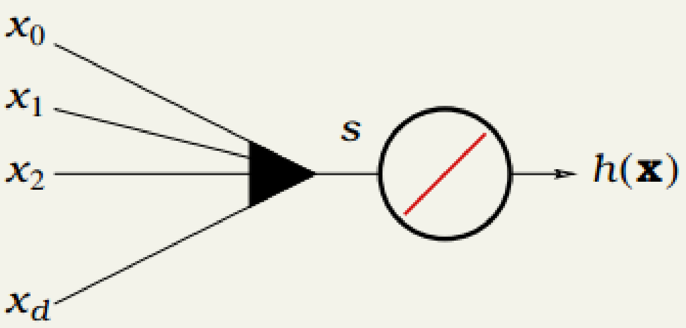
Reference Answer: 2

When $\mathbf{w}^T\mathbf{x} = 0$, $h(\mathbf{x})$ is exactly $\frac{1}{2}$. So thresholding $h(\mathbf{x})$ at $\frac{1}{2}$ is the same as thresholding $(\mathbf{w}^T\mathbf{x})$ at 0.

Linear Models

linear regression

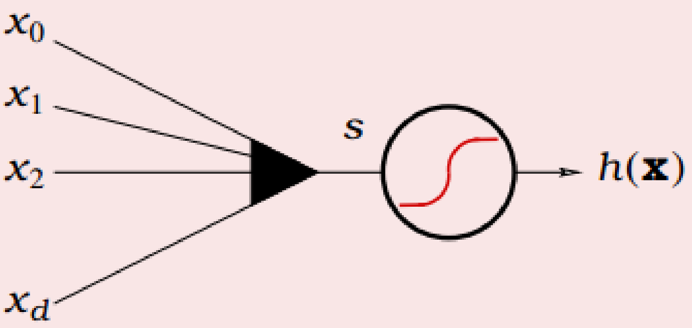
$$h(\mathbf{x}) = s$$



friendly err = squared
(easy to minimize)

logistic regression

$$h(\mathbf{x}) = z(s)$$



err = ?

How to define the cost (error) function for logistic regression?

Maximum-Likelihood Estimation

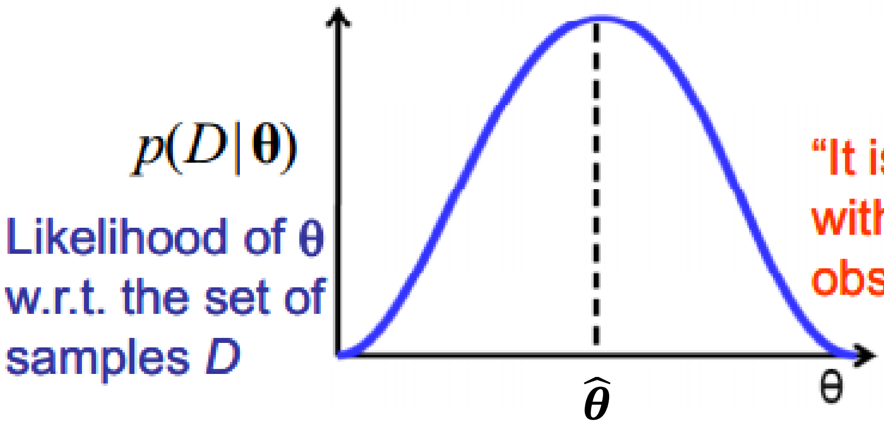
Given a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$, where the n samples are drawn independently from identical distribution $p(\boldsymbol{x}|\boldsymbol{\theta})$, estimate parameters $\boldsymbol{\theta}$.

ML estimate parameters $\boldsymbol{\theta}$ maximizes $p(\mathcal{D}|\boldsymbol{\theta})$

$\mathcal{D}$ is an i.i.d set

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta})$$

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^{n} \boldsymbol{p}(\boldsymbol{x}_k|\boldsymbol{\theta})$$



"It is the value of θ that best agrees with or supports the actually observed training samples"

Logistic Regression--$y \in \{0,1\}$

Consider $\mathcal{D} = \{(\boldsymbol{x}_1, +), (\boldsymbol{x}_2, -), \dots, (\boldsymbol{x}_m, -)\}$

Likelihood that h generates $\mathcal{D}$

$$p(\boldsymbol{x}_1)h(\boldsymbol{x}_1)$$

$$p(\boldsymbol{x}_2)(1 - h(\boldsymbol{x}_2))$$

$$\vdots$$

$$p(\boldsymbol{x}_m)(1 - h(\boldsymbol{x}_m))$$

- Target function:
  $f(x) = p(+1|\boldsymbol{x})$
- If $h \approx f$, then likelihood $(h) \approx$ that using (f)

Likelihood of Logistic Regression

Goal: $\arg\max_{h} likelihood(h)$ $\qquad$ $likelihood(h) = \prod_{i=1}^{m} p(\boldsymbol{x_i})p(y|\boldsymbol{x_i})$

Consider $\mathcal{D} = \{(\boldsymbol{x_1}, +), (\boldsymbol{x_2}, -), \dots, (\boldsymbol{x_m}, -)\}$

$$likelihood(h) = \prod_{i=1}^{m} p(\boldsymbol{x_i})p(y_i|\boldsymbol{x_i})$$

$$= p(\boldsymbol{x}_1)h(\boldsymbol{x}_1)p(\boldsymbol{x}_2)(1 - h(\boldsymbol{x}_2)) \cdots p(\boldsymbol{x}_m)(1 - h(\boldsymbol{x}_m))$$

Likelihood of Logistic Regression

Goal: $\arg\max_{h} likelihood(h)$ $\quad likelihood(h) = \prod_{i=1}^{m} p(\boldsymbol{x_i})p(y|\boldsymbol{x_i})$

Consider $\mathcal{D} = \{(\boldsymbol{x_1}, +), (\boldsymbol{x_2}, -), \dots, (\boldsymbol{x_m}, -)\}$

$$likelihood(h) = \prod_{i=1}^{m} p(\boldsymbol{x_i})p(y_i|\boldsymbol{x_i})$$

$$= p(\boldsymbol{x}_1)h(\boldsymbol{x}_1)p(\boldsymbol{x}_2)(1 - h(\boldsymbol{x}_2)) \cdots p(\boldsymbol{x}_m)(1 - h(\boldsymbol{x}_m))$$

We remove all the $p(\boldsymbol{x}_i)$ which remains the same for all the hypothesis h .

Likelihood of Logistic Regression

$$likelihood(h) = \prod_{i=1}^{m} p(\boldsymbol{x_i})p(y_i|\boldsymbol{x_i}) \propto \prod_{i=1}^{m} p(y_i|\boldsymbol{x_i})$$

$$p(y_i|\boldsymbol{x_i}) = \begin{cases} h(\boldsymbol{x_i}) & \text{for } y_i = 1 \\ 1 - h(\boldsymbol{x_i}) & \text{for } y_i = 0 \end{cases} \Longleftrightarrow p(y_i|\boldsymbol{x_i}) = h(\boldsymbol{x_i})^{y_i}(1 - h(\boldsymbol{x_i}))^{(1-y_i)}$$

Bernoulli distribution

$$likelihood(h) \propto \prod_{i=1}^{m} p(y_i|\boldsymbol{x_i}) = \prod_{i=1}^{m} h(\boldsymbol{x_i})^{y_i}(1 - h(\boldsymbol{x_i}))^{(1-y_i)}$$

Log-Likelihood of Logistic Regression

Negative Log-likelihood

$$\min_{h} E(h) = \sum_{i=1}^{m} -(y_i ln\, h(\boldsymbol{x_i}) + (1 - y_i)\, ln(1 - h(\boldsymbol{x_i}))\,)$$

Cross-entropy loss

Cross-entropy

$$H(p, q) = -\sum_{x} p(x)\log(q(x))$$

$p \in \{y, 1 - y\}$

$q \in \{h(\boldsymbol{x}), 1 - h(\boldsymbol{x})\}$

Negative Log-likelihood

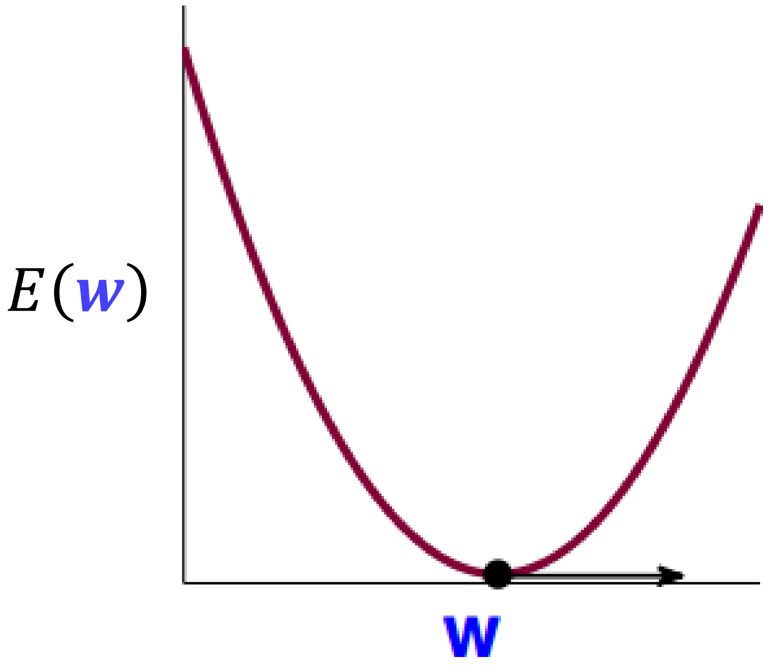
$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} \left[-y_i ln\left(\frac{1}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}_i}}\right) - (1 - y_i) ln\left(\frac{1}{1 + e^{\boldsymbol{w}^T \boldsymbol{x}_i}}\right)\right]$$

$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} \left[-y_i \boldsymbol{w}^T \boldsymbol{x_i} + ln(1 + e^{\boldsymbol{w}^T \boldsymbol{x}_i})\right]$$

Minimize $E(w)$

$$\min_{\boldsymbol{w}} E(\boldsymbol{w}) = \sum_{\mathrm{i}=1}^{m} \left[-y_i \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x_i} + ln(1 + e^{\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x}_i}) \right]$$

Cross-entropy loss



$E(w)$: continuous, differentiable, twice-differentiable, **convex**

We want to find the valley

$$\nabla E(w) = 0$$

Matrix Calculus

$$\min_{\boldsymbol{w}} E(\boldsymbol{w}) = \sum_{i=1}^{m} \left[-y_i \boldsymbol{w}^T \boldsymbol{x_i} + ln(1 + e^{\boldsymbol{w}^T \boldsymbol{x}_i})\right]$$

Identities: scalar-by-vector $\frac{\partial y}{\partial \mathbf{x}} = \nabla_{\mathbf{x}} y$

Condition	Expression	Numerator layout, i.e. by $\mathbf{x}^T$; result is row vector	Denominator layout, i.e. by $\mathbf{x}$; result is column vector
a is not a function of $\mathbf{x}$	$\frac{\partial a}{\partial \mathbf{x}} =$	$\mathbf{0}^\top$ [4]	$\mathbf{0}$ [4]
a is not a function of $\mathbf{x}$, $u = u(\mathbf{x})$	$\frac{\partial au}{\partial \mathbf{x}} =$	$a\frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$, $v = v(\mathbf{x})$	$\frac{\partial (u+v)}{\partial \mathbf{x}} =$	$\frac{\partial u}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$, $v = v(\mathbf{x})$	$\frac{\partial uv}{\partial \mathbf{x}} =$	$u\frac{\partial v}{\partial \mathbf{x}} + v\frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$	$\boxed{\frac{\partial g(u)}{\partial \mathbf{x}} =}$	$\boxed{\frac{\partial g(u)}{\partial u}\frac{\partial u}{\partial \mathbf{x}}}$	
$u = u(\mathbf{x})$	$\frac{\partial f(g(u))}{\partial \mathbf{x}} =$	$\frac{\partial f(g)}{\partial g}\frac{\partial g(u)}{\partial u}\frac{\partial u}{\partial \mathbf{x}}$	
$\mathbf{u} = \mathbf{u}(\mathbf{x})$, $\mathbf{v} = \mathbf{v}(\mathbf{x})$	$\frac{\partial(\mathbf{u}\cdot\mathbf{v})}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}^\top \mathbf{v}}{\partial \mathbf{x}} =$	$\mathbf{u}^\top\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^\top\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ • assumes numerator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{v} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{u}$ • assumes denominator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$

Gradient $\nabla E(w)$

$$\nabla E(\boldsymbol{w}) = \sum_{\mathrm{i}=1}^{m} \left[-y_i \boldsymbol{x}_i + \frac{e^{\boldsymbol{w}^T \boldsymbol{x}_i}}{1 + e^{\boldsymbol{w}^T \boldsymbol{x}_i}} \boldsymbol{x}_i \right] = \sum_{\mathrm{i}=1}^{m} \left[z\left(\boldsymbol{w}^T \boldsymbol{x}_i\right) - y_i \right] \boldsymbol{x}_i = 0$$

- $\nabla E(\boldsymbol{w})$ is a non-linear equation of $\boldsymbol{w}$
  - It is hard to derive the closed form solution. :-(

Gradient $\nabla E(w)$

$$\nabla E(\boldsymbol{w}) = \sum_{\mathrm{i}=1}^{m}\left[-y_i\boldsymbol{x_i} + \frac{e^{\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_i}}{1+e^{\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x}_i}}\boldsymbol{x_i}\right] = \sum_{\mathrm{i}=1}^{m}\left[z\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x_i}\right) - y_i\right]\boldsymbol{x_i} = 0$$

- Apply the iterative optimization to the logistic regression.

Iterative Optimization

Optimization Methods

- Optimization: either minimize or maximize some function $f(x)$ by altering x.
- In most cases, optimization refers to the minimization of $f(x)$.

Maximization $f(x)$ $\Leftrightarrow$ **Minimization** $-f(x)$

- $f(x)$: objective function, cost function, loss function, error function.
- The value that minimize $f(x)$: $x^* = \arg\min f(x)$.

Optimization Methods

- **Deterministic Optimization**
  - The data for the given problem are known accurately.
- **Stochastic Optimization**
  - Refers to a collection of methods for minimizing or maximizing an objective function when randomness is present.

Deterministic Optimization

- First-order methods: methods that use only the gradient.
- Second-order methods: methods that also use the Hessian matrix.

$$\boldsymbol{H}(f)_{i,j} = \frac{\partial^2}{\partial x_i \partial x_j} f(\boldsymbol{x})$$

$\boldsymbol{x}$: multiple input dimensions.

Taylor Approximation

Expansion at x_0

$$f(x) = \frac{f(x_0)}{0!} + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n + R_n(x)$$

Examples

$$e^x = 1 + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + o\left(x^3\right)$$

$$\ln(1 + x) = x - \frac{1}{2}x^2 + \frac{1}{3}x^3 + o\left(x^3\right)$$

$$\sin x = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 + o\left(x^5\right)$$

Gradient Descent [Cauchy 1847]

- Motivation: to minimize the local first-order Taylor approximation of f

$$\min_x f(x) \approx \min_x f(x_t) + \nabla f(x_t)^T (x - x_t)$$

- Update rule:

$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

Where $\eta_t > 0$ is the step-size (learning rate).

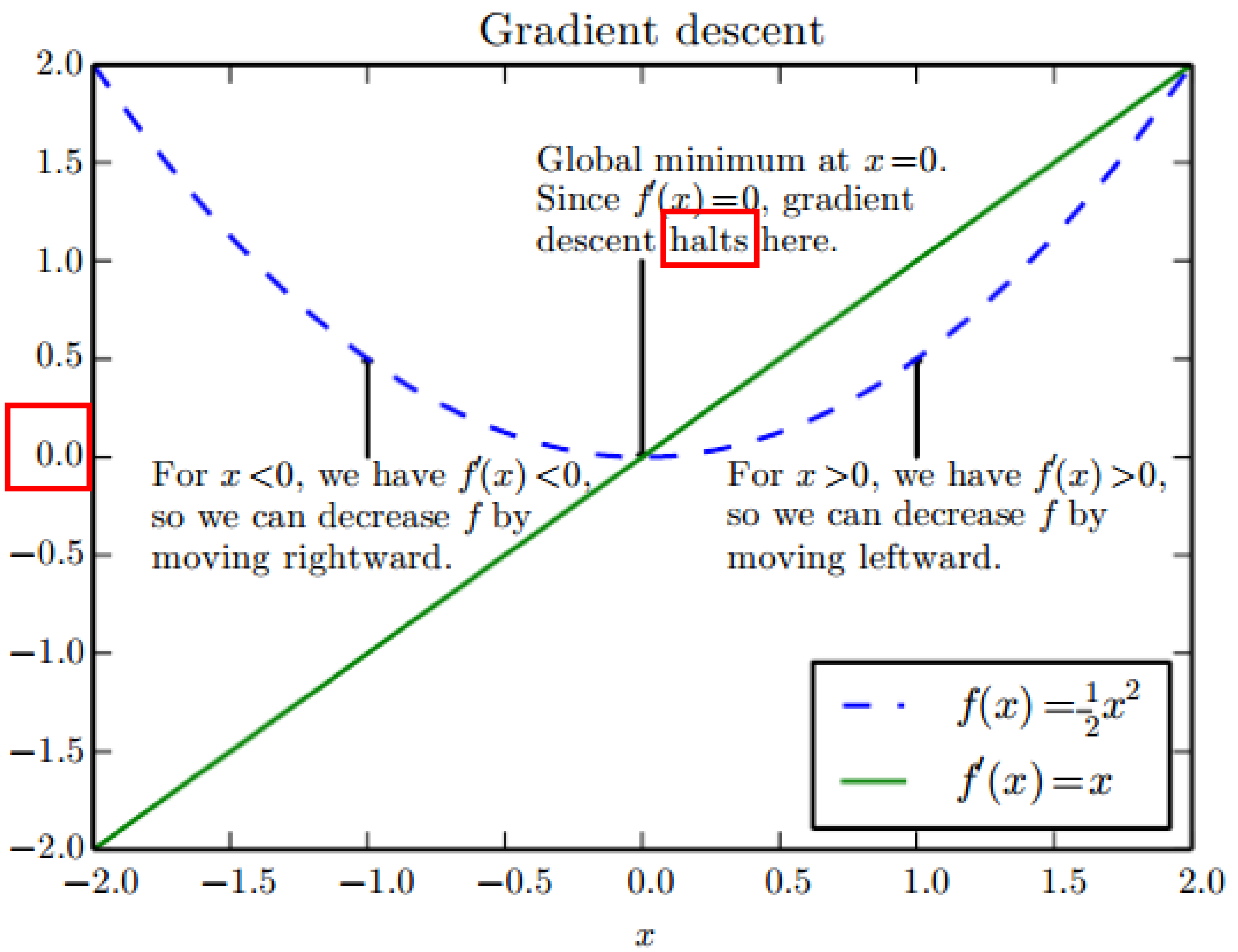
Interpretation

- Reduce $f(x)$ by moving x in small steps with opposite sign of the derivative.

- Update rule:

$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

- Critical/stationary points: Points where $f'(x) = 0$ 驻点



An illustration of gradient descent.

Interpretation

- At each iteration, consider the expansion

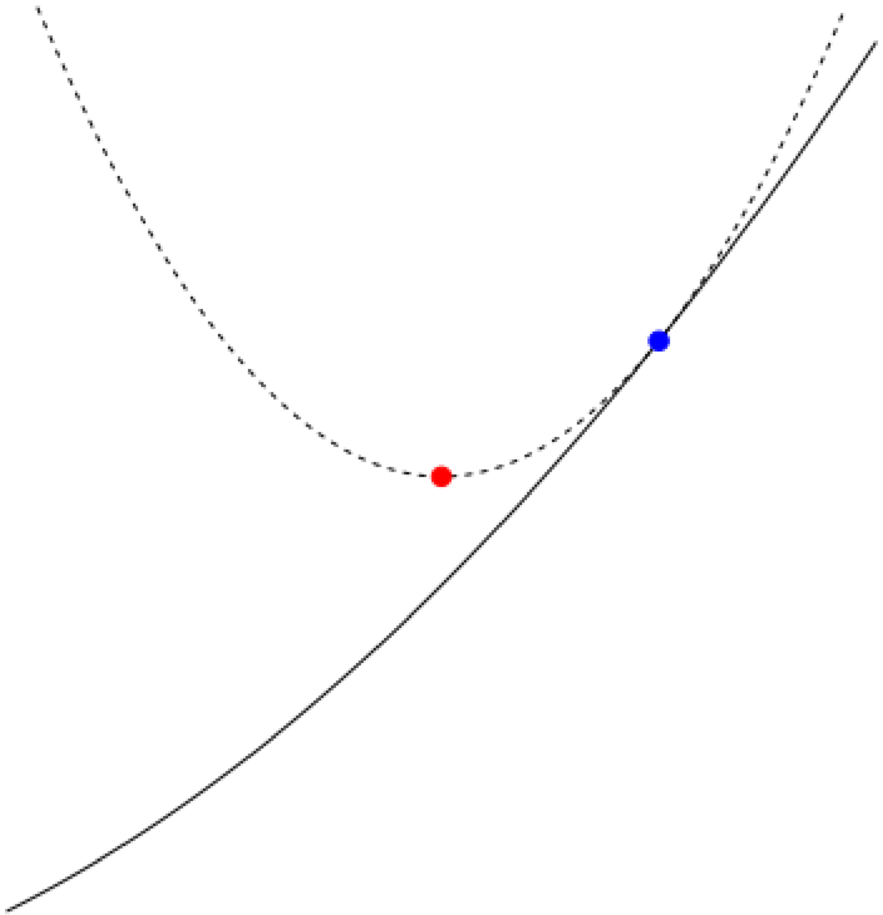
$$f(x) \approx \underbrace{f(x_t) + \nabla f(x_t)^T (x - x_t)}_{\text{Linear approximation of } f} + \underbrace{\frac{1}{2\eta_t} \|x - x_t\|^2}_{\text{Proximity term with weight } \frac{1}{2\eta_t}}$$

- Quadratic approximation, replacing usual $\nabla^2 f(x)$ by $\frac{1}{\eta_t} I$:

$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

Interpretation

$$f(x) \approx f(x_t) + \nabla f(x_t)^T\,(x - x_t) + \frac{1}{2\eta_t}\|x - x_t\|^2$$

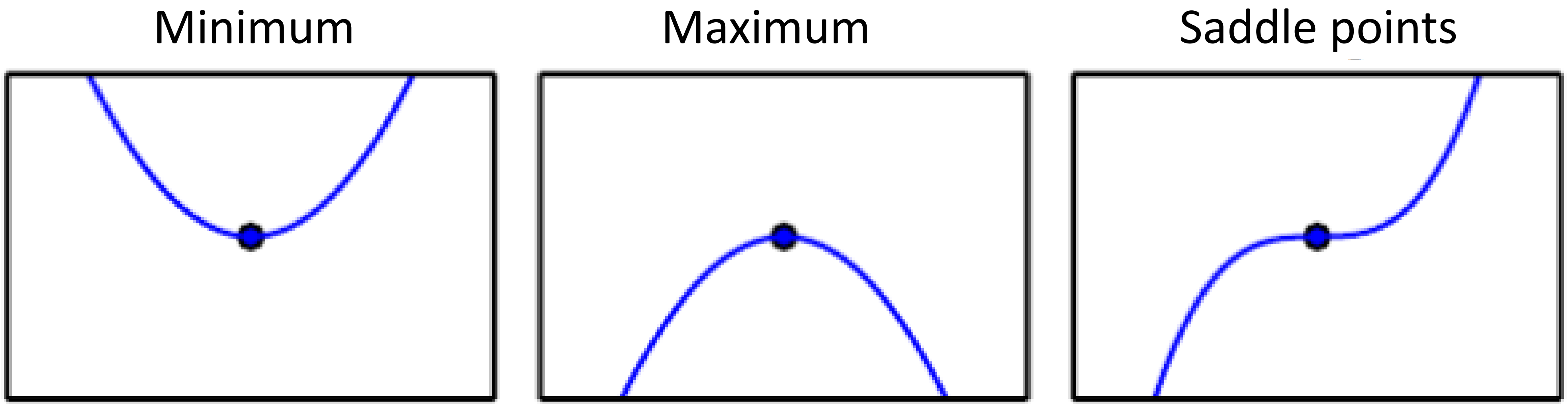


Blue point is x_t, red point is x_{t+1}.

Global VS Local Minimum

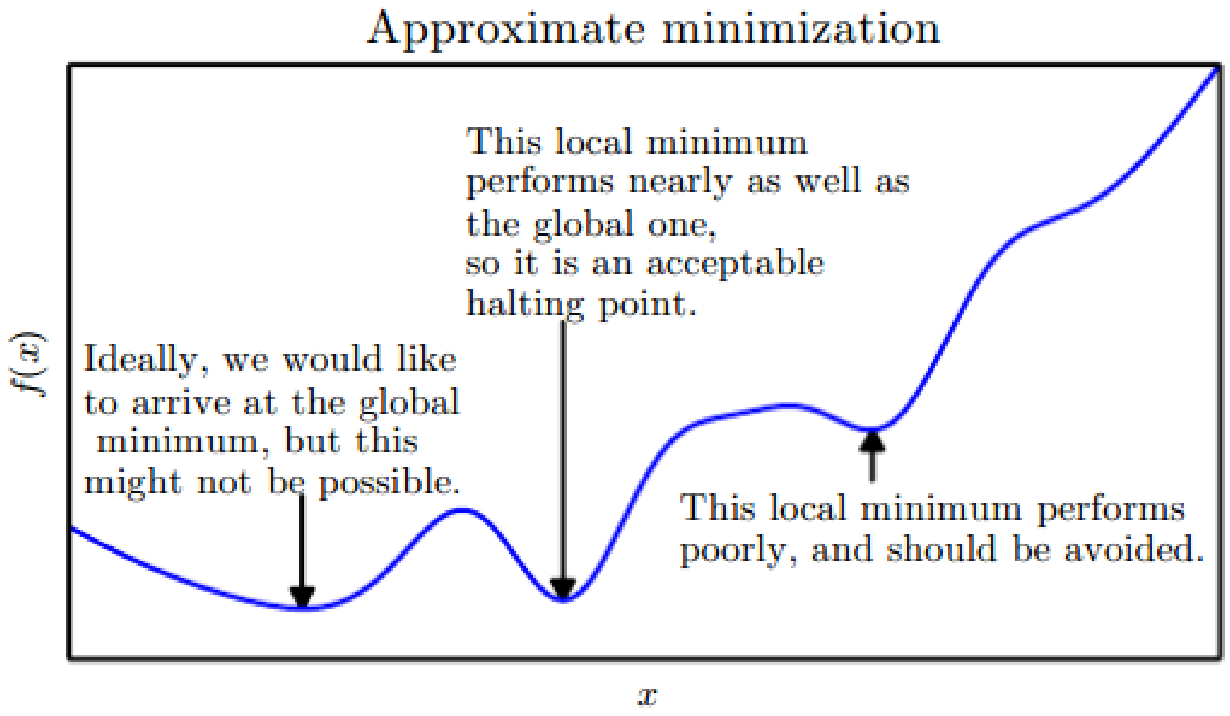
- Global minimum: a point that obtains the absolute lowest value of $f(x)$.
- Local minimum: a point where $f(x)$ is lower than at all neighboring points.
- Saddle points: some critical points are neither maxima or minima. 鞍点

Types of critical points



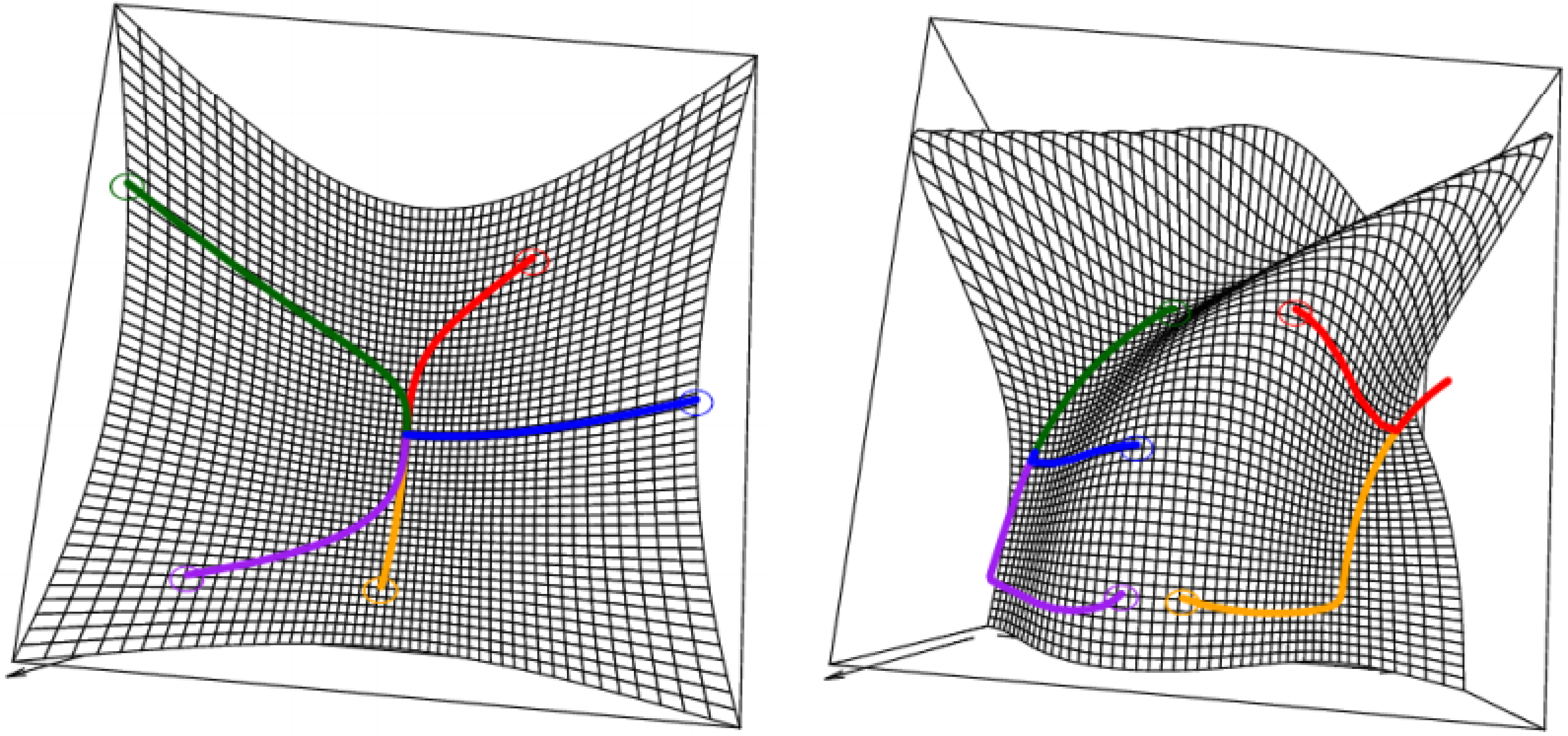
Global VS Local Minimum

- Global minimum: a point that obtains the absolute lowest value of $f(x)$.
- Local minimum: a point where $f(x)$ is higher than at all neighboring points.
- Saddle points: some critical points are neither maxima or minima. 鞍点



Different Starting Points

- Gradient Descent with different starting points are illustrated in different colors.



(a) Convex function (b) Non-convex function

- (a): Strictly convex function: Converge to the global optimum.
- (b): Non-convex function: Different paths may end up at different local optima.

Gradient Descent [Cauchy 1847]

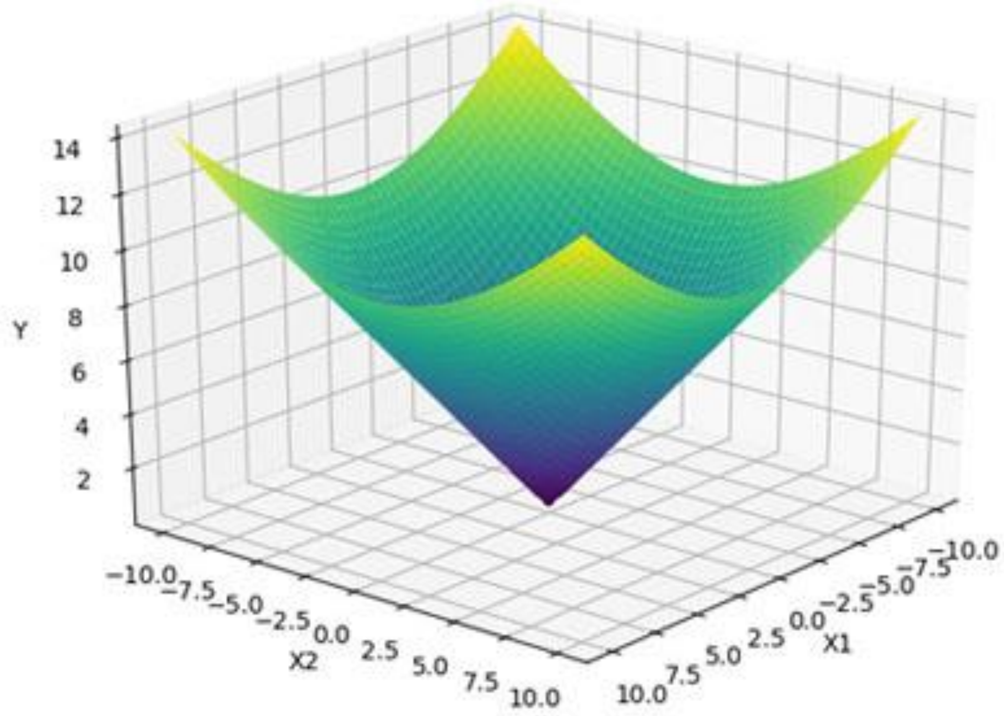
$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

- Gradient Descent requires a step size η controlling the amount of gradient updated to the current point at each iteration.
- It is naïve to set $\eta_t = \eta$ for all iterations.

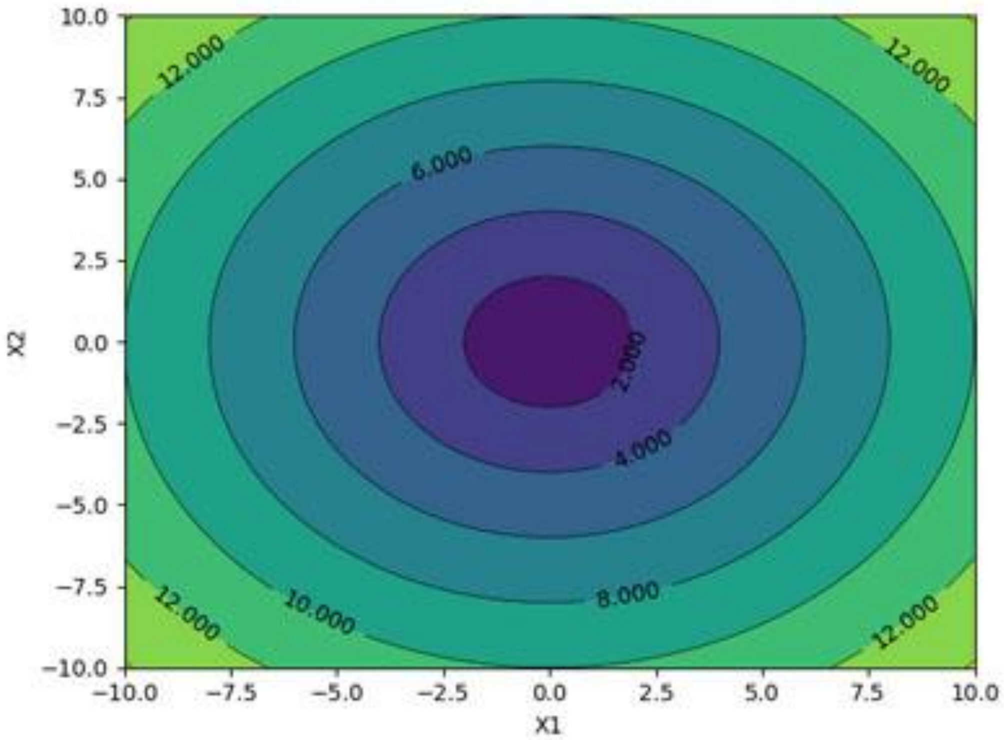
How to choose step sizes?

Fixed Step Sizes

Considering $f(x) = (10x_1^2 + x_2^2)/2$



3D Plot



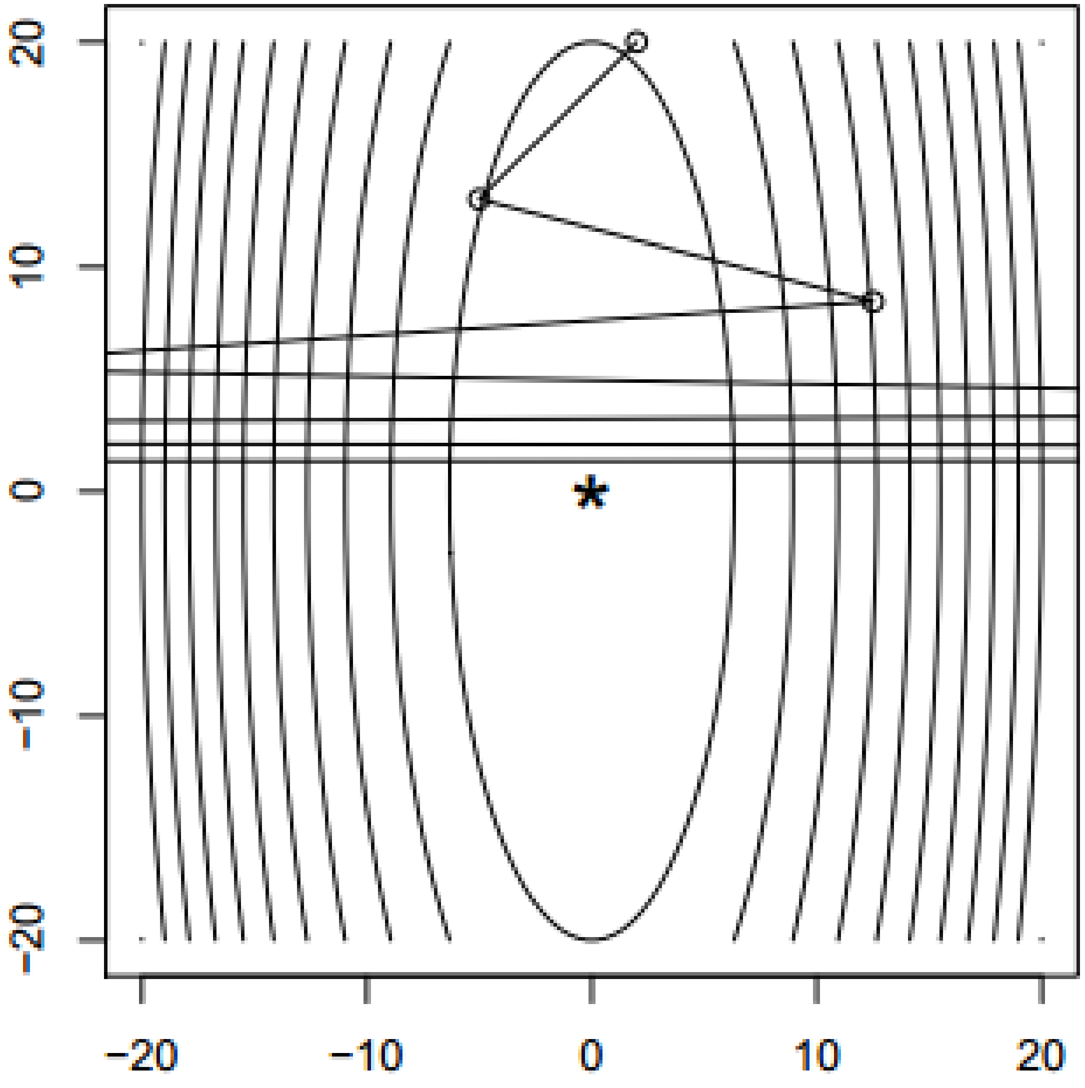
Contour Plot

Fixed Step Sizes

Considering $f(x) = (10x_1^2 + x_2^2)/2$

If η is too big, can lead to divergence.

- The learning function oscillates away from the optimal point.
- As shown, it oscillates after 8 steps.

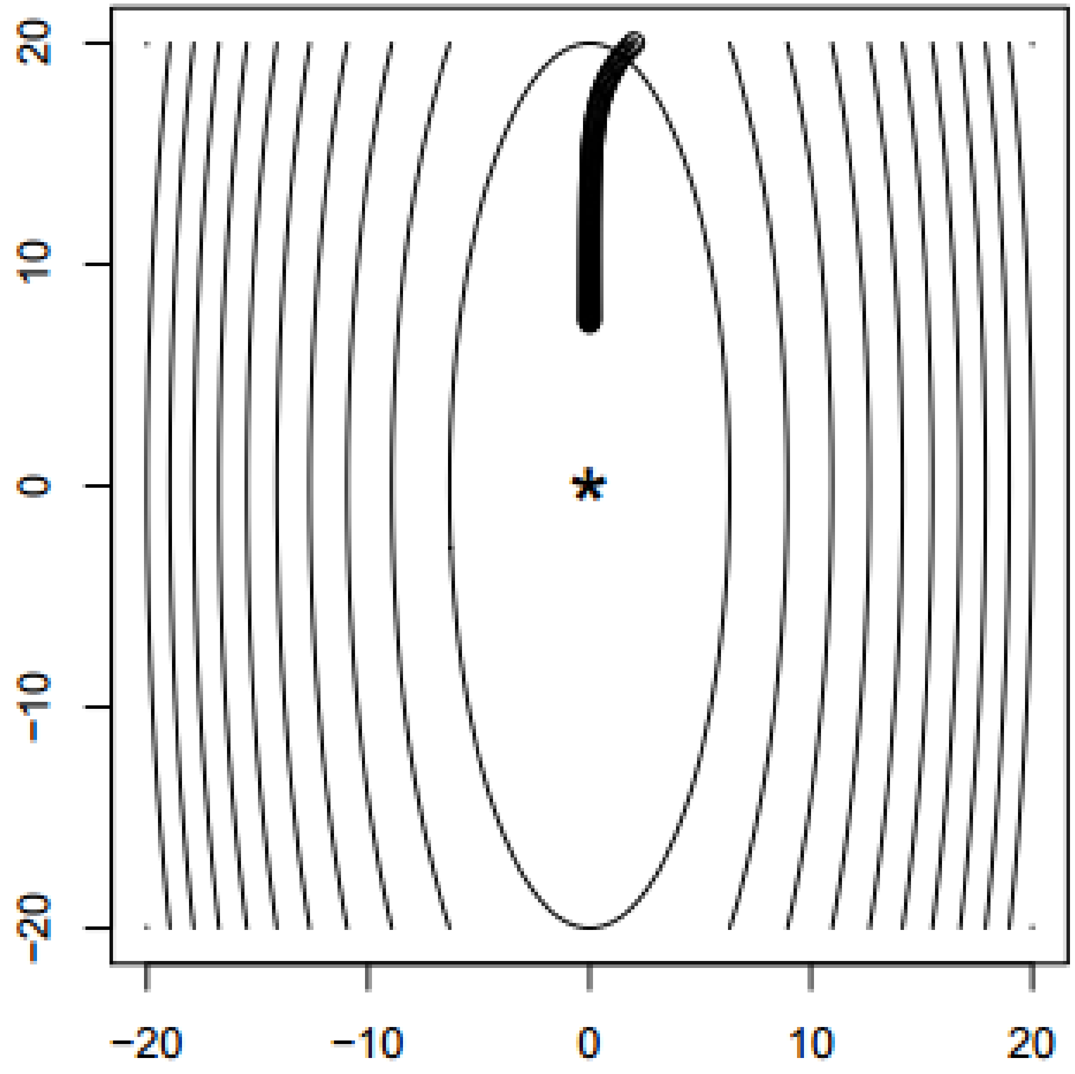


Fixed Step Sizes

Considering $f(x) = (10x_1^2 + x_2^2)/2$

If η is too small, takes longer time for the function to converge.

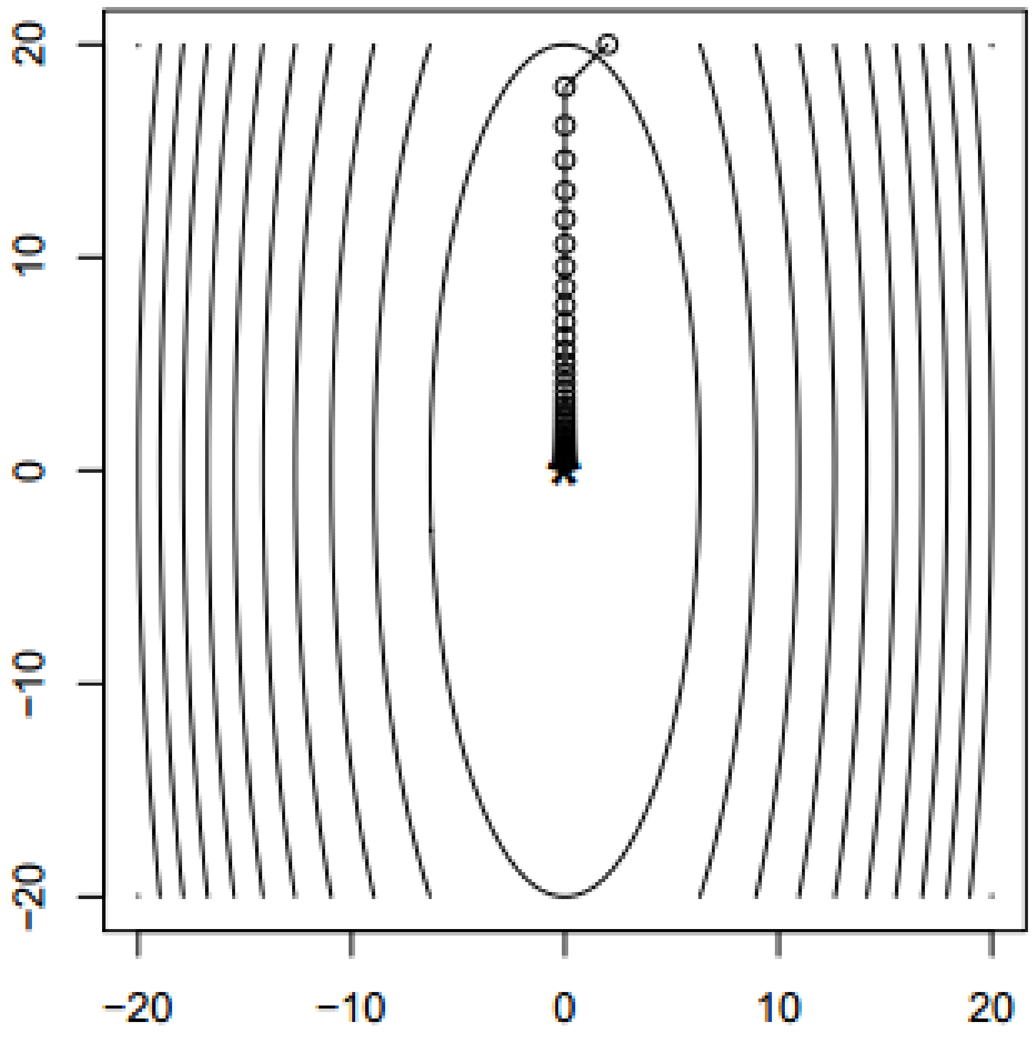
- As shown, GD after 100 steps.



Fixed Step Sizes

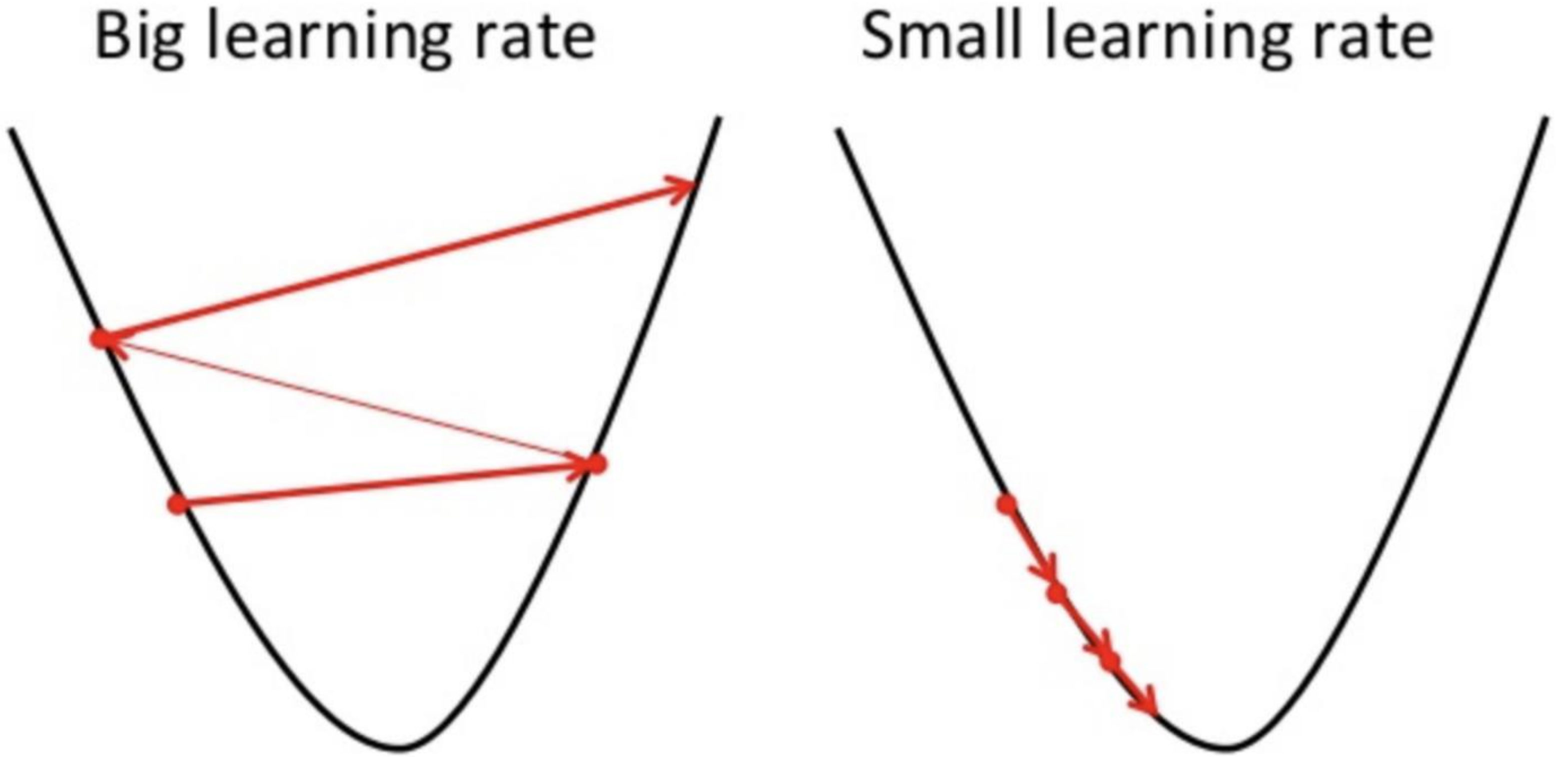
Considering $f(x) = (10x_1^2 + x_2^2)/2$

Same example, gradient descent after 40 appropriately sized steps.



Fixed Step Sizes

Considering $f(x) = x^2/2$



Deterministic Optimization

- First-order methods: methods that use only the gradient.
- Second-order methods: methods that also use the Hessian matrix.

Suppose $f: \mathbb{R}^n \to \mathbb{R}$ is a function taking as input a vector $x \in \mathbb{R}^n$ and outputting a scalar $f(\boldsymbol{x}) \in \mathbb{R}$; if all second partial derivatives of f exist and are continuous over the domain of the function, then the Hessian matrix $\boldsymbol{H}$ of f is a square $n \times n$ matrix, usually defined as follows.

$$H = \nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

, or

$$H_{ij} = \frac{\partial^2 f}{\partial x_i x_j}$$

Newton's Methods

- Motivation: to minimize the local second-order Taylor approximation of f.

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \approx \min_{\boldsymbol{x}} f(\boldsymbol{x}_t) + \nabla f(\boldsymbol{x}_t)^T(\boldsymbol{x} - \boldsymbol{x}_t) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_t)^T \nabla^2 f(\boldsymbol{x}_t)(\boldsymbol{x} - \boldsymbol{x}_t)$$

- Take the derivative of $\boldsymbol{x}$ on both side, we have,

$$\frac{df(\boldsymbol{x})}{d\boldsymbol{x}} = \nabla f(x_t) + \nabla^2 f(\boldsymbol{x}_t)(\boldsymbol{x} - \boldsymbol{x}_t) = \mathbf{0}$$

- Update rule: suppose $\nabla^2 f(x_t)$ is positive definite,

$$\boldsymbol{x} = \boldsymbol{x}_t - [\nabla^2 f(x_t)]^{-1} \nabla f(x_t)$$

Newton's Methods

- Motivation: to minimize the local second-order Taylor approximation of f.

$$\min_x f(x) \approx \min_x f(x_t) + f'(x_t)(x - x_t) + \frac{1}{2} f''(x_t)(x - x_t)^2$$

- Take the derivative of x on both side, we have,

$$f'(x) = f'(x_t) + f''(x_t)(x - x_t) = 0$$

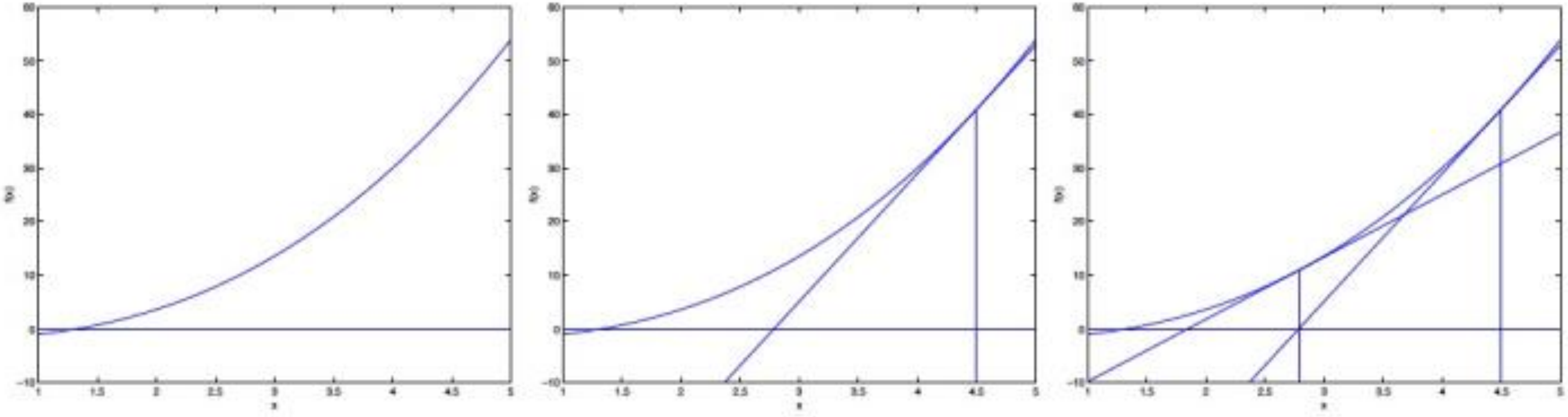
- Update rule: suppose $f''(x_t) \neq 0$,

$$x = x_t - \frac{f'(x_t)}{f''(x_t)}$$

Newton’s Methods

- In numerical analysis, Newton’s Methods is to find successively better approximations to the roots of a real-valued function, $(\mathrm{i.e}, f(z) = 0)$.

$$z = z_t - \frac{f(z_t)}{f'(z_t)}$$



- In optimization, we want to find the stationary point $f'(x_t) = 0$, i.e.,

$$x = x_t - \frac{f'(x_t)}{f''(x_t)}$$

Newton’s Methods

- **Advantage:**
  - More accurate local approximation of the objective,
  - The convergence is much faster.
- **Disadvantage:**
  - Need to compute the second derivatives
  - Need to compute the inverse of Hessian (time/storage consuming)

Go back to logistic regression

Gradient $\nabla E(w)$

$$\nabla E(\boldsymbol{w}) = \sum_{i=1}^{m}\left[-y_i\boldsymbol{x_i} + \frac{e^{\boldsymbol{w}^T\boldsymbol{x_i}}}{1+e^{\boldsymbol{w}^T\boldsymbol{x_i}}}\boldsymbol{x_i}\right] = \sum_{i=1}^{m}[z(\boldsymbol{w}^T\boldsymbol{x_i}) - y_i]\boldsymbol{x_i} = \boldsymbol{X}^T(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{md} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_m^T \end{pmatrix} \in \mathbb{R}^{m\times(d+1)},\ \hat{\boldsymbol{y}} = \begin{pmatrix} z(\boldsymbol{w}^T\boldsymbol{x}_1) \\ z(\boldsymbol{w}^T\boldsymbol{x}_2) \\ \vdots \\ z(\boldsymbol{w}^T\boldsymbol{x}_m) \end{pmatrix} \in \mathbb{R}^m,\ \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m$$

- Apply the Newton's method to the logistic regression,

$$\boldsymbol{x} = \boldsymbol{x}_t - [\nabla^2 f(x_t)]^{-1}\nabla f(x_t) \quad \Rightarrow \quad \boldsymbol{w} = \boldsymbol{w}_t - \boldsymbol{H}(\boldsymbol{w}_t)^{-1}\nabla E(\boldsymbol{w}_t)$$

- Need to solve,

$$\boldsymbol{H} = \nabla^2 E(\boldsymbol{w}) = \frac{\nabla E(\boldsymbol{w})}{\nabla \boldsymbol{w}} = ?$$

Gradient $\nabla E(w)$

$$\nabla E(\boldsymbol{w}) = \sum_{i=1}^{m}[z(\boldsymbol{w}^T\boldsymbol{x}_i) - y_i]\boldsymbol{x}_i$$

$$\boldsymbol{H} = \nabla^2 E(\boldsymbol{w}) = \frac{\nabla E(\boldsymbol{w})}{\nabla \boldsymbol{w}}$$

$$\boldsymbol{H} = \sum_{i=1}^{m}\frac{\nabla\{z(\boldsymbol{w}^T\boldsymbol{x}_i)\boldsymbol{x}_i\}}{\nabla \boldsymbol{w}}$$

Identities: vector-by-vector $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$

Condition	Expression	Numerator layout, i.e. by $\mathbf{y}$ and $\mathbf{x}^T$	Denominator layout, i.e. by $\mathbf{y}^T$ and $\mathbf{x}$
a is not a function of **x**	$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} =$	$\mathbf{0}$	
	$\frac{\partial \mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{I}$	
A is not a function of **x**	$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{A}$	$\mathbf{A}^\top$
A is not a function of **x**	$\frac{\partial \mathbf{x}^\top \mathbf{A}}{\partial \mathbf{x}} =$	$\mathbf{A}^\top$	$\mathbf{A}$
a is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	
a = *a*(**x**), **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{u}\frac{\partial a}{\partial \mathbf{x}}$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial a}{\partial \mathbf{x}}\mathbf{u}^\top$
A is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial \mathbf{A}\mathbf{u}}{\partial \mathbf{x}} =$	$\mathbf{A}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{A}^\top$
u = **u**(**x**), **v** = **v**(**x**)	$\frac{\partial(\mathbf{u}+\mathbf{v})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	
u = **u**(**x**)	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}$
u = **u**(**x**)	$\frac{\partial \mathbf{f}(\mathbf{g}(\mathbf{u}))}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}$

Gradient $\nabla E(w)$

$$\nabla E(\boldsymbol{w}) = \sum_{i=1}^{m}\left[z\left(\boldsymbol{w}^T\boldsymbol{x_i}\right) - y_i\right]\boldsymbol{x_i}$$

$$\boldsymbol{H} = \nabla^2 E(\boldsymbol{w}) = \frac{\nabla E(\boldsymbol{w})}{\nabla \boldsymbol{w}}$$

$$\boldsymbol{H} = \sum_{i=1}^{m}\frac{\nabla\{z\left(\boldsymbol{w}^T\boldsymbol{x_i}\right)\boldsymbol{x_i}\}}{\nabla \boldsymbol{w}}$$

a: $z\left(\boldsymbol{w}^T\boldsymbol{x_i}\right)$

$\boldsymbol{u}(\boldsymbol{w})$: $\boldsymbol{x_i}$

$\frac{\nabla z\left(\boldsymbol{w}^T\boldsymbol{x_i}\right)}{\nabla \boldsymbol{w}}$ is a scalar –by-vector problem.

Identities: vector-by-vector $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$

Condition	Expression	Numerator layout, i.e. by y and x^T	Denominator layout, i.e. by y^T and x
a is not a function of **x**	$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} =$	$\mathbf{0}$	
	$\frac{\partial \mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{I}$	
A is not a function of **x**	$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{A}$	$\mathbf{A}^\top$
A is not a function of **x**	$\frac{\partial \mathbf{x}^\top\mathbf{A}}{\partial \mathbf{x}} =$	$\mathbf{A}^\top$	$\mathbf{A}$
a is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	
a = *a*(**x**), **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{u}\frac{\partial a}{\partial \mathbf{x}}$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial a}{\partial \mathbf{x}}\mathbf{u}^\top$
A is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial \mathbf{A}\mathbf{u}}{\partial \mathbf{x}} =$	$\mathbf{A}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{A}^\top$
u = **u**(**x**), **v** = **v**(**x**)	$\frac{\partial (\mathbf{u}+\mathbf{v})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	
u = **u**(**x**)	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}$
u = **u**(**x**)	$\frac{\partial \mathbf{f}(\mathbf{g}(\mathbf{u}))}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}$

Gradient $\nabla E(w)$

$\frac{\nabla z(w^T x_i)}{\nabla w}$ is a scalar –by-vector problem

Identities: scalar-by-vector $\frac{\partial y}{\partial \mathbf{x}} = \nabla_{\mathbf{x}} y$

Condition	Expression	Numerator layout, i.e. by $\mathbf{x}^\mathrm{T}$; result is row vector	Denominator layout, i.e. by $\mathbf{x}$; result is column vector
a is not a function of $\mathbf{x}$	$\frac{\partial a}{\partial \mathbf{x}} =$	$\mathbf{0}^\top$ [4]	$\mathbf{0}$ [4]
a is not a function of $\mathbf{x}$, $u = u(\mathbf{x})$	$\frac{\partial au}{\partial \mathbf{x}} =$	$a\frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$, $v = v(\mathbf{x})$	$\frac{\partial (u+v)}{\partial \mathbf{x}} =$	$\frac{\partial u}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$, $v = v(\mathbf{x})$	$\frac{\partial uv}{\partial \mathbf{x}} =$	$u\frac{\partial v}{\partial \mathbf{x}} + v\frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$	$\frac{\partial g(u)}{\partial \mathbf{x}} =$	$\frac{\partial g(u)}{\partial u}\frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$	$\frac{\partial f(g(u))}{\partial \mathbf{x}} =$	$\frac{\partial f(g)}{\partial g}\frac{\partial g(u)}{\partial u}\frac{\partial u}{\partial \mathbf{x}}$	
$\mathbf{u} = \mathbf{u}(\mathbf{x})$, $\mathbf{v} = \mathbf{v}(\mathbf{x})$	$\frac{\partial (\mathbf{u}\cdot\mathbf{v})}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}^\top\mathbf{v}}{\partial \mathbf{x}} =$	$\mathbf{u}^\top\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^\top\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ • assumes numerator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{v} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{u}$ • assumes denominator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$

Gradient $\nabla E(w)$

$\frac{\nabla z(\boldsymbol{w}^T \boldsymbol{x}_i)}{\nabla \boldsymbol{w}}$ is a scalar –by-vector problem $u: \boldsymbol{w}^T \boldsymbol{x}_i$ $z: g$ $\frac{\nabla z(\boldsymbol{w}^T \boldsymbol{x}_i)}{\nabla \boldsymbol{w}} = z(\boldsymbol{w}^T \boldsymbol{x}_i) z(-\boldsymbol{w}^T \boldsymbol{x}_i) \boldsymbol{x}_i$

Identities: scalar-by-vector $\frac{\partial y}{\partial \mathbf{x}} = \nabla_{\mathbf{x}} y$

Condition	Expression	Numerator layout, i.e. by $\mathbf{x}^T$; result is row vector	Denominator layout, i.e. by $\mathbf{x}$; result is column vector
a is not a function of $\mathbf{x}$	$\frac{\partial a}{\partial \mathbf{x}} =$	$\mathbf{0}^\top$ [4]	$\mathbf{0}$ [4]
a is not a function of $\mathbf{x}$, $u = u(\mathbf{x})$	$\frac{\partial au}{\partial \mathbf{x}} =$	$a \frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$, $v = v(\mathbf{x})$	$\frac{\partial (u+v)}{\partial \mathbf{x}} =$	$\frac{\partial u}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$, $v = v(\mathbf{x})$	$\frac{\partial uv}{\partial \mathbf{x}} =$	$u \frac{\partial v}{\partial \mathbf{x}} + v \frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$	$\frac{\partial g(u)}{\partial \mathbf{x}} =$	$\frac{\partial g(u)}{\partial u} \frac{\partial u}{\partial \mathbf{x}}$	
$u = u(\mathbf{x})$	$\frac{\partial f(g(u))}{\partial \mathbf{x}} =$	$\frac{\partial f(g)}{\partial g} \frac{\partial g(u)}{\partial u} \frac{\partial u}{\partial \mathbf{x}}$	
$\mathbf{u} = \mathbf{u}(\mathbf{x})$, $\mathbf{v} = \mathbf{v}(\mathbf{x})$	$\frac{\partial (\mathbf{u} \cdot \mathbf{v})}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}^\top \mathbf{v}}{\partial \mathbf{x}} =$	$\mathbf{u}^\top \frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^\top \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ • assumes numerator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \mathbf{v} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \mathbf{u}$ • assumes denominator layout of $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$

Gradient $\nabla E(w)$

$$\nabla E(\mathbf{w}) = \sum_{i=1}^{m}\left[z(\mathbf{w}^T\mathbf{x}_i) - y_i\right]\mathbf{x}_i$$

$$\mathbf{H} = \nabla^2 E(\mathbf{w}) = \frac{\nabla E(\mathbf{w})}{\nabla \mathbf{w}}$$

$$\mathbf{H} = \sum_{i=1}^{m} \frac{\nabla\{z(\mathbf{w}^T\mathbf{x}_i)\mathbf{x}_i\}}{\nabla \mathbf{w}}$$

a: $z(\mathbf{w}^T\mathbf{x}_i)$

$\mathbf{u}(\mathbf{w})$: $\mathbf{x}_i$

$$\frac{\nabla z(\mathbf{w}^T\mathbf{x}_i)}{\nabla \mathbf{w}} = z(\mathbf{w}^T\mathbf{x}_i)\,z(-\mathbf{w}^T\mathbf{x}_i)\,\mathbf{x}_i$$

$$\mathbf{H} = \sum_{i=1}^{m} \mathbf{x}_i z(\mathbf{w}^T\mathbf{x}_i)\,z(-\mathbf{w}^T\mathbf{x}_i)\,\mathbf{x}_i^T$$

Identities: vector-by-vector $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$

Condition	Expression	Numerator layout, i.e. by y and $\mathbf{x}^\mathsf{T}$	Denominator layout, i.e. by $\mathbf{y}^\mathsf{T}$ and x
a is not a function of **x**	$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} =$	$\mathbf{0}$	
	$\frac{\partial \mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{I}$	
A is not a function of **x**	$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$	$\mathbf{A}$	$\mathbf{A}^\top$
A is not a function of **x**	$\frac{\partial \mathbf{x}^\top\mathbf{A}}{\partial \mathbf{x}} =$	$\mathbf{A}^\top$	$\mathbf{A}$
a is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	
a = *a*(**x**), **u** = **u**(**x**)	$\frac{\partial a\mathbf{u}}{\partial \mathbf{x}} =$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{u}\frac{\partial a}{\partial \mathbf{x}}$	$a\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial a}{\partial \mathbf{x}}\mathbf{u}^\top$
A is not a function of **x**, **u** = **u**(**x**)	$\frac{\partial \mathbf{A}\mathbf{u}}{\partial \mathbf{x}} =$	$\mathbf{A}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{A}^\top$
u = **u**(**x**), **v** = **v**(**x**)	$\frac{\partial(\mathbf{u}+\mathbf{v})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	
u = **u**(**x**)	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}$
u = **u**(**x**)	$\frac{\partial \mathbf{f}(\mathbf{g}(\mathbf{u}))}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\frac{\partial \mathbf{g}(\mathbf{u})}{\partial \mathbf{u}}\frac{\partial \mathbf{f}(\mathbf{g})}{\partial \mathbf{g}}$

Gradient $\nabla E(w)$

$$\nabla E(\boldsymbol{w}) = \sum_{i=1}^{m}\left[-y_i\boldsymbol{x_i} + \frac{e^{\boldsymbol{w}^T\boldsymbol{x}_i}}{1+e^{\boldsymbol{w}^T\boldsymbol{x}_i}}\boldsymbol{x_i}\right] = \sum_{i=1}^{m}\left[z(\boldsymbol{w}^T\boldsymbol{x_i}) - y_i\right]\boldsymbol{x_i} = \boldsymbol{X}^{\boldsymbol{T}}(\hat{\boldsymbol{y}} - \boldsymbol{y})$$

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{m1} & x_{m2} & \cdots & x_{md} \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\boldsymbol{T}} \\ \boldsymbol{x}_2^{\boldsymbol{T}} \\ \vdots \\ \boldsymbol{x}_m^{\boldsymbol{T}} \end{pmatrix} \in \mathbb{R}^{m\times(d+1)}, \hat{\boldsymbol{y}} = \begin{pmatrix} z(\boldsymbol{w}^T\boldsymbol{x}_1) \\ z(\boldsymbol{w}^T\boldsymbol{x}_2) \\ \vdots \\ z(\boldsymbol{w}^T\boldsymbol{x}_m) \end{pmatrix} \in \mathbb{R}^m, \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m$$

$$\boldsymbol{H} = \nabla^2 E(\boldsymbol{w}) = \frac{\nabla E(\boldsymbol{w})}{\nabla \boldsymbol{w}} = \sum_{i=1}^{m}\boldsymbol{x_i}z(\boldsymbol{w}^T\boldsymbol{x_i})z(-\boldsymbol{w}^T\boldsymbol{x_i})\boldsymbol{x}_i^{\boldsymbol{T}} = \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{X}$$

$\boldsymbol{R} \in \mathbb{R}^{m\times m}$ is a diagonal matrix with elements $R_{ii} = z(\boldsymbol{w}^T\boldsymbol{x_i})z(-\boldsymbol{w}^T\boldsymbol{x_i})$

- Apply the Newton's method to the logistic regression,

$$\boldsymbol{w} = \boldsymbol{w}_t - \boldsymbol{H}(\boldsymbol{w}_t)^{-1}\nabla E(\boldsymbol{w}_t)$$

Compare with Linear Regression

For the linear regression with the sum-of-squares error function, we have,

$$E(\boldsymbol{w}) = \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 = (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T\,(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

$$\nabla E(\boldsymbol{w}) = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y}$$

$$\boldsymbol{H} = \nabla^2 E(\boldsymbol{w}) = \frac{\nabla E(\boldsymbol{w})}{\nabla \boldsymbol{w}} = \boldsymbol{X}^T\boldsymbol{X}$$

$\boldsymbol{H}$ is a constant: the error function is quadratic.

Apply the Newton's method to the linear regression,

$$\boldsymbol{w} = \boldsymbol{w}_t - \boldsymbol{H}(\boldsymbol{w}_t)^{-1}\nabla E(\boldsymbol{w}_t)$$

$$\boldsymbol{w} = \boldsymbol{w}_t - (\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w}_t - \boldsymbol{X}^T\boldsymbol{y}) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$ Closed-form

The Newton method gives the exact solution in one step.

Summary

Linear Regression

- **Problem**
  - Use hyperplanes to approximate real values
- **Error (Cost) function**
  - Least square
  - $E(\boldsymbol{w})$: continuous, differentiable, **convex**
- **Algorithm**
  - Analytic solution with pseudo-inverse

Summary

Logistic Regression

- **Problem**
  - $P(+1|x)$ as target and $z(\boldsymbol{w}^T \boldsymbol{x}_i)$ as hypotheses
- **Error (Cost) Function**
  - Negative log-likelihood (cross-entropy)
  - $E(\boldsymbol{w})$: continuous, differentiable, twice-differentiable, **convex**
- **Optimization**
  - Iterative methods, e.g., Gradient descent, Newton’s method

Exercise

$y \in \{0,1\}$

Target function:
$f(x) = p(+1|\boldsymbol{x})$

$\Longleftrightarrow$

$$p(y|\boldsymbol{x}) = \begin{cases} h(\boldsymbol{x}) & \text{for } y = 1 \\ 1 - h(\boldsymbol{x}) & \text{for } y = 0 \end{cases}$$

$y \in \{-1,1\}$

Target function:
$f(x) = p(+1|\boldsymbol{x})$

$\Longleftrightarrow$

$$p(y|\boldsymbol{x}) = \begin{cases} h(\boldsymbol{x}) & \text{for } y = 1 \\ 1 - h(\boldsymbol{x}) & \text{for } y = -1 \end{cases}$$

Can you derive the objective function?

Logistic Regression--$y \in \{-1,1\}$

Consider $\mathcal{D} = \{(\boldsymbol{x_1}, +), (\boldsymbol{x_2}, -), \dots, (\boldsymbol{x_m}, -)\}$

$$h(\boldsymbol{x}_i) = P(+1|\boldsymbol{x}_i) \quad \Leftrightarrow \quad p(y|\boldsymbol{x}_i) = \begin{cases} h(\boldsymbol{x}_i) & \text{for } y = +1 \\ 1 - h(\boldsymbol{x}_i) & \text{for } y = -1 \end{cases}$$

$$\Leftrightarrow \quad p(y|\boldsymbol{x}_i) = \begin{cases} h(\boldsymbol{x}_i) & \text{for } y = +1 \\ h(-\boldsymbol{x}_i) & \text{for } y = -1 \end{cases} \quad \Leftrightarrow \quad p(y|\boldsymbol{x}_i) = h(y\boldsymbol{x}_i)$$

$$\boxed{1 - z(s) = z(-s)}$$

$$\boxed{z(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}}}$$

Logistic Regression--$y \in \{-1,1\}$

Consider $\mathcal{D} = \{(\boldsymbol{x_1}, +), (\boldsymbol{x_2}, -), \dots, (\boldsymbol{x_m}, -)\}$

$$likelihood(h) = \prod_{i=1}^{m} p(\boldsymbol{x_i})p(y_i|\boldsymbol{x_i}) = p(\boldsymbol{x_1})h(\boldsymbol{x_1})p(\boldsymbol{x_2})h(-\boldsymbol{x_2}) \dots p(\boldsymbol{x_m})h(-\boldsymbol{x_m})$$

$$\max_h likelihood(h) \propto \prod_{i=1}^{m} p(y_i|\boldsymbol{x_i}) = \prod_{i=1}^{m} h(y_i\boldsymbol{x_i}) = \prod_{i=1}^{m} \theta(y_i\boldsymbol{w}^T\boldsymbol{x_i})$$

$$\min_{\boldsymbol{w}} -\sum_{i=1}^{m} \ln \theta(y_i\boldsymbol{w}^T\boldsymbol{x_i}) \quad \Leftrightarrow \quad \min_{\boldsymbol{w}} -\sum_{i=1}^{m} \ln 1/(1 + e^{-y_i\boldsymbol{w}^T\boldsymbol{x}_i})$$

Cross-entropy loss for $y \in \{-1,1\}$

$$\boxed{\min_{\boldsymbol{w}} -\frac{1+y_i}{2}\sum_{i=1}^{m} \ln \frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}_i}} - \frac{1-y_i}{2}\sum_{i=1}^{m} \ln \frac{1}{1+e^{\boldsymbol{w}^T\boldsymbol{x}_i}}}$$

Logistic Regression--$y \in \{-1,1\}$

Cross-entropy

$$H(p,q) = -\sum_{x} p(x)\log(q(x))$$

$$p \in \left\{\frac{1+y_i}{2}, \frac{1-y_i}{2}\right\}$$

$$q \in \{h(\boldsymbol{x}), 1-h(\boldsymbol{x})\}$$

$$\min_{\boldsymbol{w}} -\frac{1+y_i}{2}\sum_{\mathrm{i}=1}^{m} \ln\frac{1}{1+e^{-\boldsymbol{w}^T\boldsymbol{x}_i}} - \frac{1-y_i}{2}\sum_{\mathrm{i}=1}^{m} \ln\frac{1}{1+e^{\boldsymbol{w}^T\boldsymbol{x}_i}}$$

Simplified function

$$\min_{\boldsymbol{w}} -\sum_{\mathrm{i}=1}^{m} \ln 1/(1+e^{-y_i\boldsymbol{w}^T\boldsymbol{x}_i})$$