Machine Learning & Pattern Recognition

SONG Xuemeng

sxmustc@gmail.com

http://xuemeng.bitcron.com/

Linear Regression

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

Training dataset: $\mathcal{D} = \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \dots, (\boldsymbol{x_m}, y_m)\}$;
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: $f(\boldsymbol{x_i}) = \boldsymbol{w^T x_i} + b = \sum_{j=1}^{d} w_j x_{ij} + b$, where $\boldsymbol{w} = (w_1\ w_2\ \dots\ w_d)^{\boldsymbol{T}} \in \mathbb{R}^{\boldsymbol{d}}$

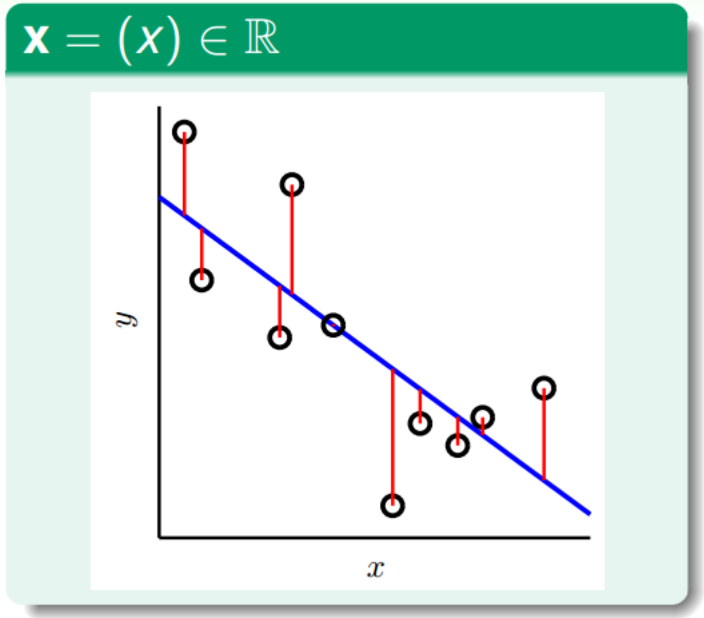
For simplicity, the bias b can be merged into the weight $\boldsymbol{w}$:

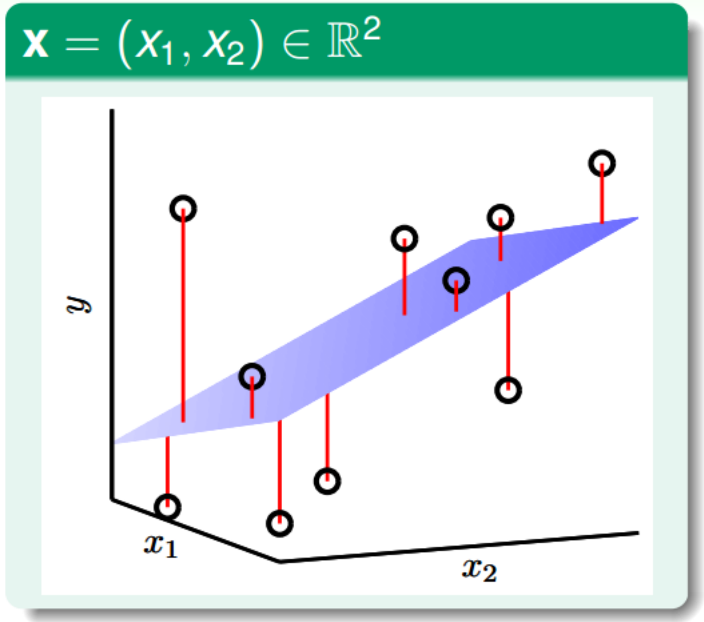
$$h(\boldsymbol{x_i}) = \boldsymbol{\widehat{w}^T \widehat{x}_i}$$

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

Linear Regression

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

Empirical Error

We usually 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 $err(\hat{y} - y) = (\hat{y} - y)^2$

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

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

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

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

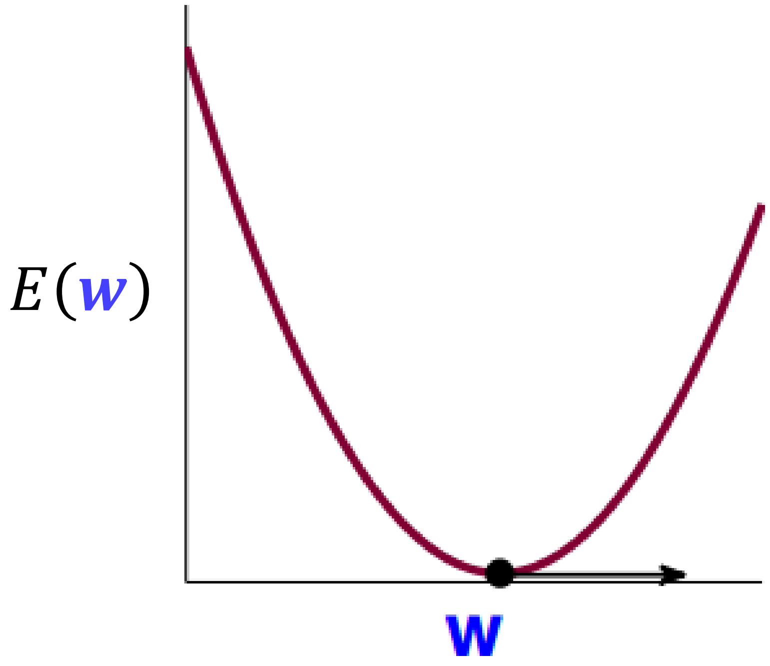
$$= \left\| \begin{matrix} \boldsymbol{x_1}^T \boldsymbol{w} - y_1 \\ \boldsymbol{x_2}^T \boldsymbol{w} - y_2 \\ \vdots \\ \boldsymbol{x_m}^T \boldsymbol{w} - y_m \end{matrix} \right\|^2 = \left\| \begin{bmatrix} - - \boldsymbol{x_1}^T - - \\ - - \boldsymbol{x_2}^T - - \\ \vdots \\ - - \boldsymbol{x_m}^T - - \end{bmatrix} \boldsymbol{w} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \right\|^2$$

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

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

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

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



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

Invertible/positive definite $X^T X$

- Unique solution

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

- Often the case because $N \gg d + 1$

Singular $X^T X$

- Many optimal solutions
- One of the solution
  - Define $\boldsymbol{X}^\dagger$ in other ways

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 $\boldsymbol{X}^\dagger$ routine**

Exercise

After getting w^*, we can calculate the predictions $\hat{y}_n = (w^*)^T x_n$. If all $\hat{y}_n$ are collected in a vector $\hat{\mathbf{y}}$ similar to how we form $\mathbf{y}$, what is the matrix formula of $\hat{\mathbf{y}}$?

1. $\mathbf{y}$
2. $\mathrm{X}\mathrm{X}^T\mathbf{y}$
3. $\mathrm{X}\mathrm{X}^\dagger\mathbf{y}$
4. $\mathrm{X}\mathrm{X}^\dagger\mathrm{X}\mathrm{X}^T\mathbf{y}$

Exercise

After getting w^* , we can calculate the predictions $\hat{y}_n = (w^*)^T x_n$. If all $\hat{y}_n$ are collected in a vector $\hat{\mathbf{y}}$ similar to how we form $\mathbf{y}$, what is the matrix formula of $\hat{\mathbf{y}}$?

1. $\mathbf{y}$
2. $XX^T\mathbf{y}$
3. $XX^\dagger\mathbf{y}$
4. $XX^\dagger XX^T\mathbf{y}$

Reference Answer: 3

Note that $\hat{\mathbf{y}} = Xw^*$. Then, a simple substitution of w^* reveals the answer.

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

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

Actual data

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

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

Logistic Hypothesis

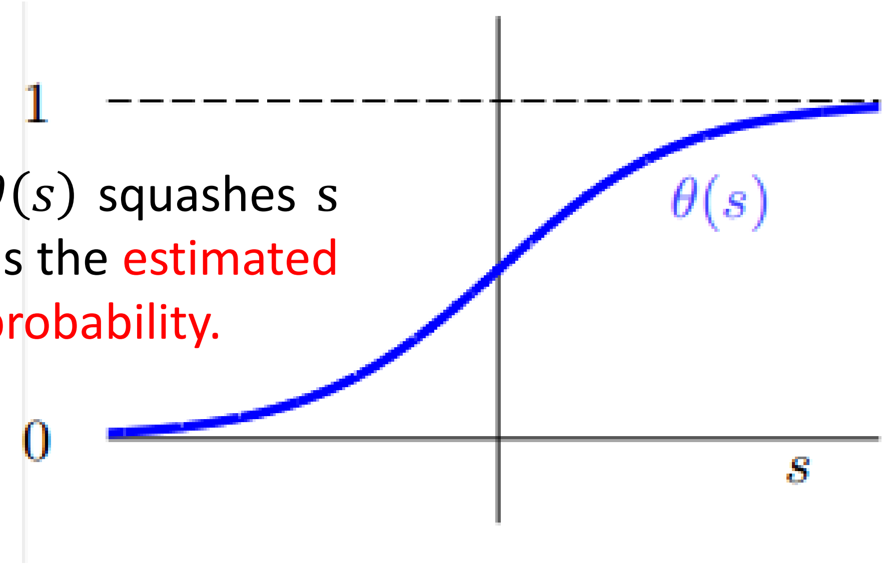
age	40 years
gender	male
blood pressure	130/85
cholesterol level	240

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

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

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

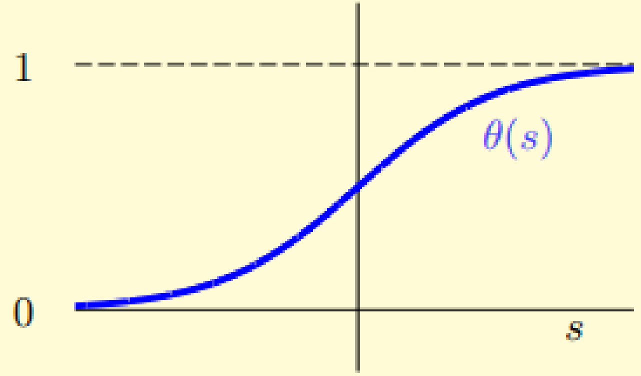
$\theta(s)$ squashes s as the estimated probability.



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

Logistic Function

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



smooth, monotonic, sigmoid function of s

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

Logistic regression use $h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$ 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

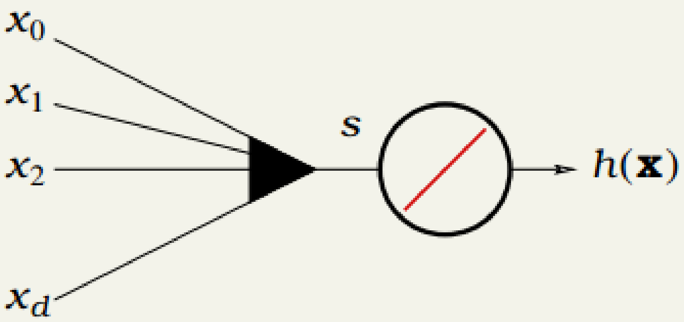
Reference Answer: 2

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

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

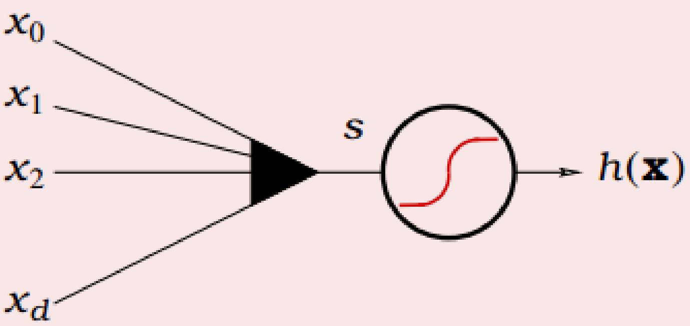


friendly err = squared

(easy to minimize)

logistic regression

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



err = ?

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

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

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

$\Longleftrightarrow$

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

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

Likelihood that h generates $\mathcal{D}$

Maximum-Likelihood Estimation

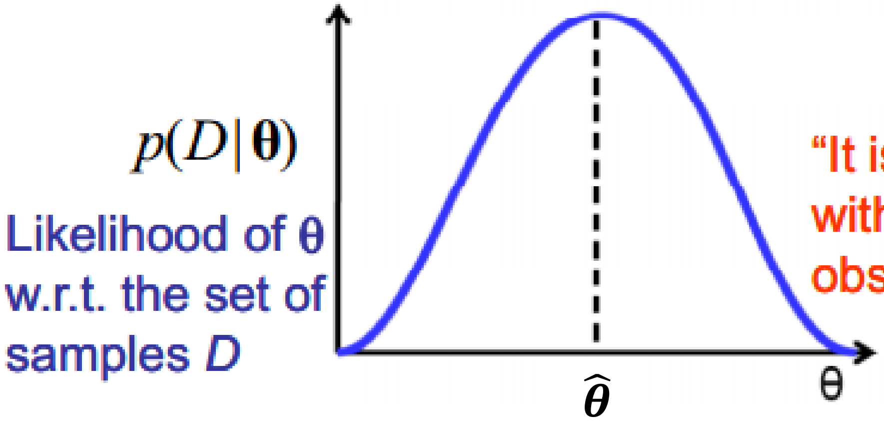
Given a dataset $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$, where the n samples are drawn independently from identical distribution $\mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta})$, estimate parameters $\boldsymbol{\theta}$.

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

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

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

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



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

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

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

$\Longleftrightarrow$

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

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

Likelihood that h generates $\mathcal{D}$

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

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

$$\vdots$$

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

- If $h \approx f$, then likelihood $(h) \approx$ that using (f)
- Probability using (f) is usually large

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

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

$$= p(x_1)h(\boldsymbol{x}_1)p(x_2)(1 - h(\boldsymbol{x}_2))\, p(x_m)(1 - h(\boldsymbol{x}_m))$$

We remove all the $p(x_i)$ which remains the same for all the hypothesis h .

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

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

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

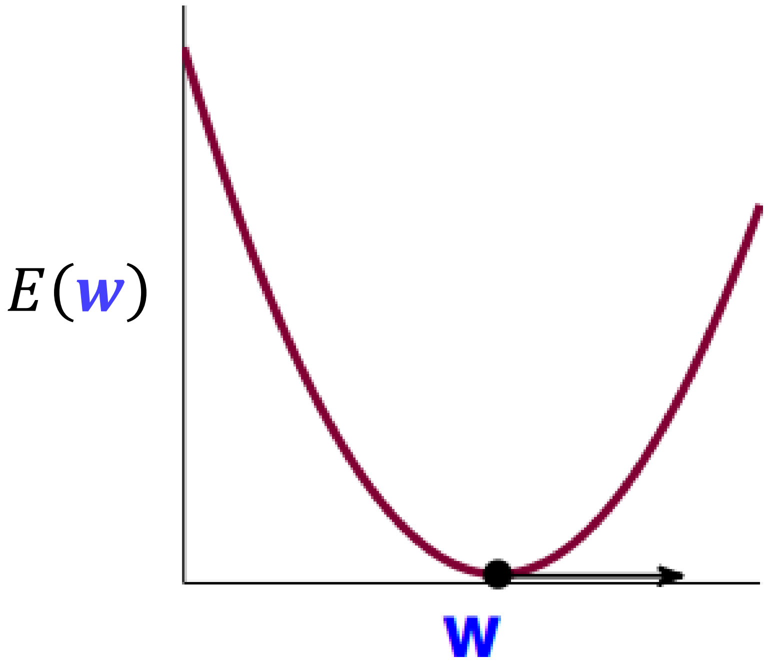
$$\min_{\boldsymbol{w}} \sum_{i=1}^{m} \left[-y_i ln\left(\frac{1}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}_i}}\right) - (1 - y_i) ln\left(\frac{1}{1 + e^{\boldsymbol{w}^T \boldsymbol{x}_i}}\right)\right]$$

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

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

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

Gradient $\nabla E(w)$

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

- $\nabla E(\boldsymbol{w}) = 0 \Leftrightarrow \begin{cases}\theta(\boldsymbol{w}^T\boldsymbol{x}_i) = 1, \text{if } y_i = 1\\ \theta(\boldsymbol{w}^T\boldsymbol{x}_i) = 0, \text{if } y_i = 0\end{cases} \Leftrightarrow \begin{cases}\boldsymbol{w}^T\boldsymbol{x}_i \to \infty, \text{if } y_i = 1\\ \boldsymbol{w}^T\boldsymbol{x}_i \to -\infty, \text{if } y_i = 0\end{cases}$
  - ➢ The data must be linearly separable. :-(
- $\nabla E(\boldsymbol{w})$ is a non-linear equation of $\boldsymbol{w}$
  - ➢ It is hard to derive the closed form solution. :-(

Gradient Descent [Cauchy 1847]

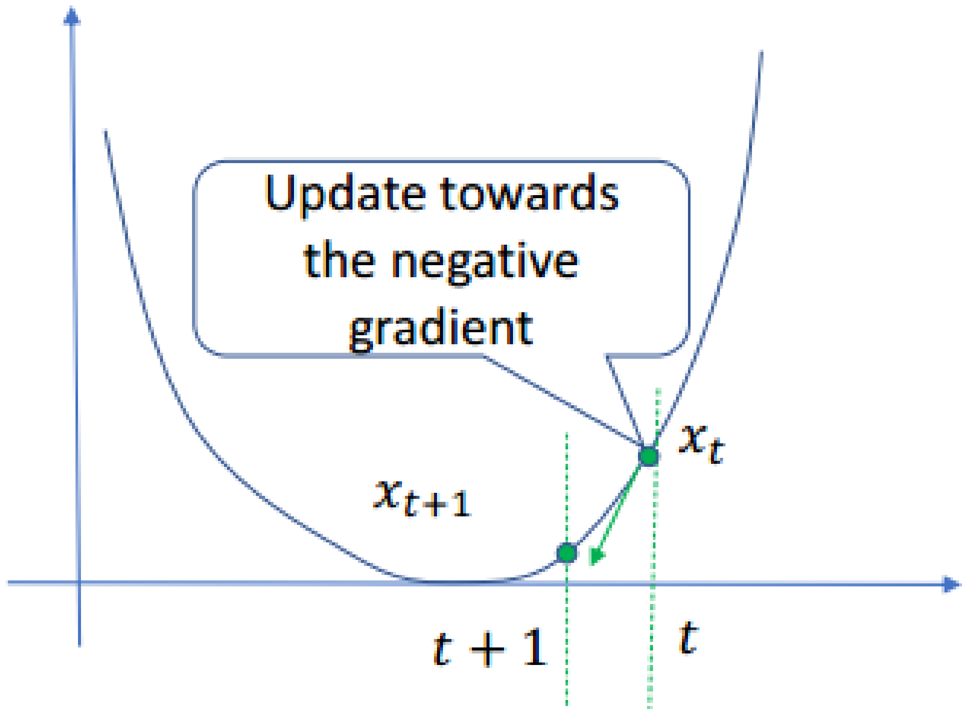
- Motivation: to minimize the local first-order Taylor approximation of f

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

- Update rule:

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

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



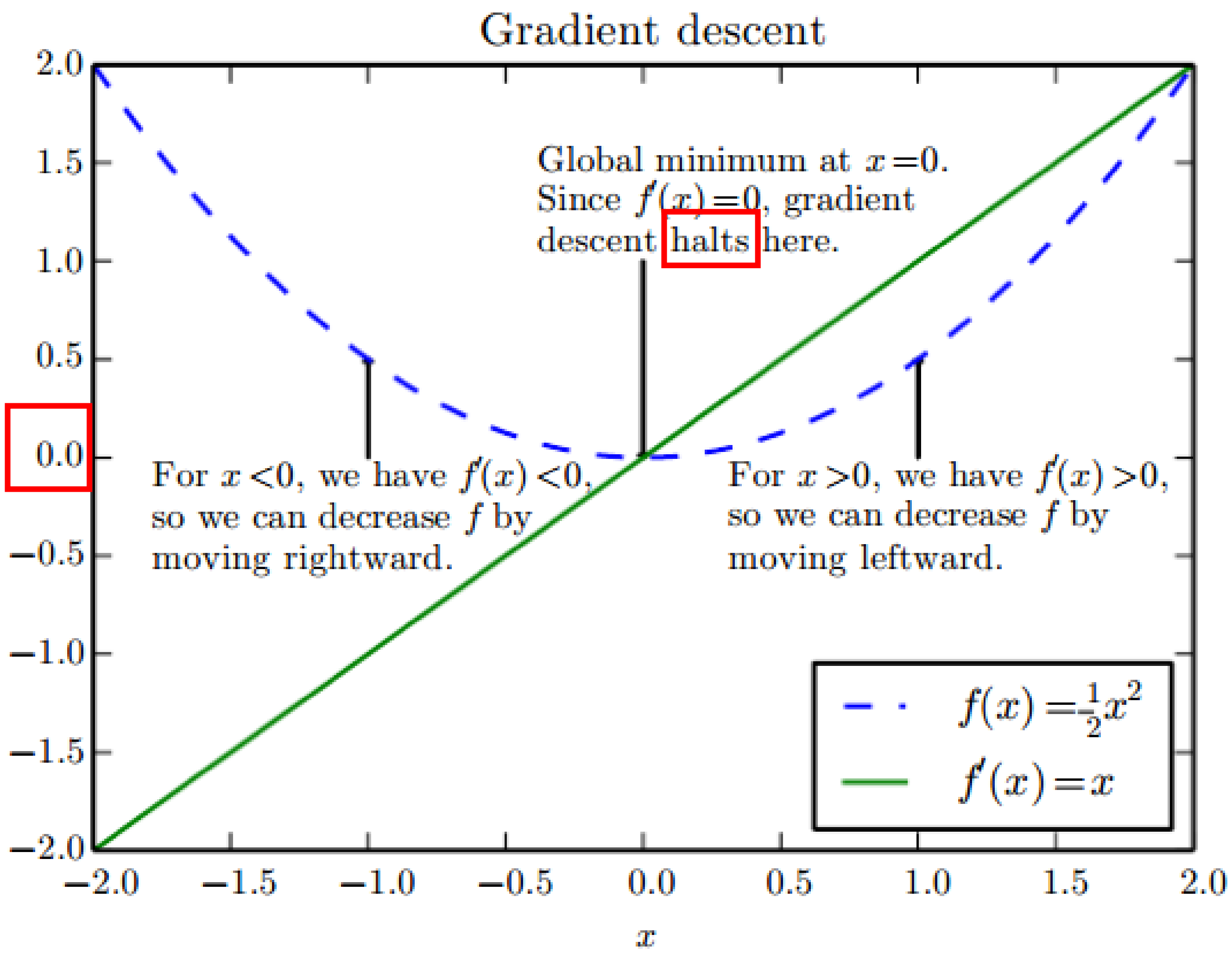
Gradient Descent--Interpretation

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



An illustration of gradient descent.