MAC 0460 / 5832
Introduction to Machine Learning

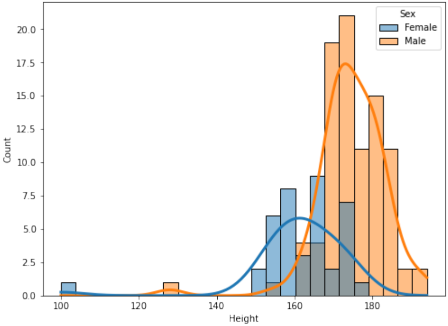
05 – Logistic regression

• binary classification • target distribution •
• likelihood function • cross-entropy loss •

IME/USP

Classification

Suppose we know the height of a person. Can we guess correctly if this person is Female or Male ?



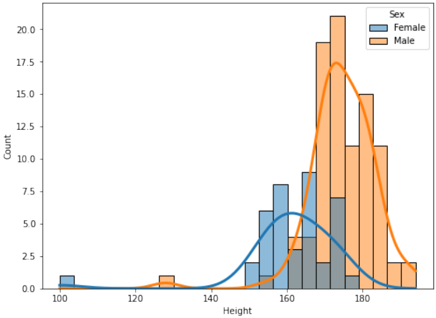
Statistical approach

Bayes' Theorem

$$P(y|\mathbf{x}) = \frac{P(y)p(\mathbf{x}|y)}{p(\mathbf{x})}$$

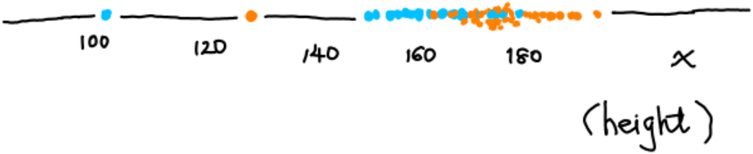
If you know the distributions, you have the winning rule:

$$y^* = \arg\max_y \{P(y|\mathbf{x})\}$$



We do not have the distribution

We only have the observations



How to solve the problem?

$\Longrightarrow$ What about the PERCEPTRON algorithm ?

No, it works only if classes are linearly separable ...

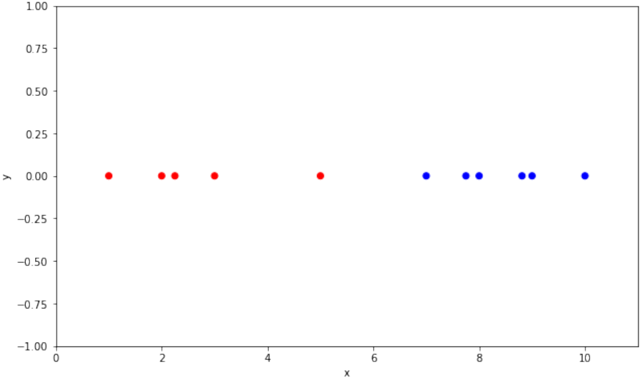
$\Longrightarrow$ We could employ the POCKET version of the PERCEPTRON

No, there must be something else ...

$\Longrightarrow$ We could employ linear regression ? Let's see!

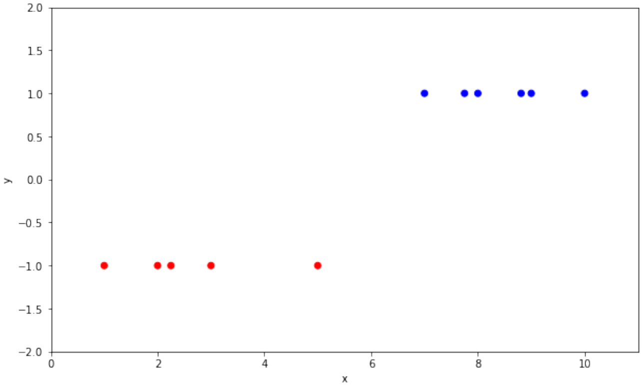
Example: $D_X = \{1, 2, 2.25, 3, 5, 7, 7.75, 8, 8.81, 9, 10\}$

Negatives: red Positives: blue



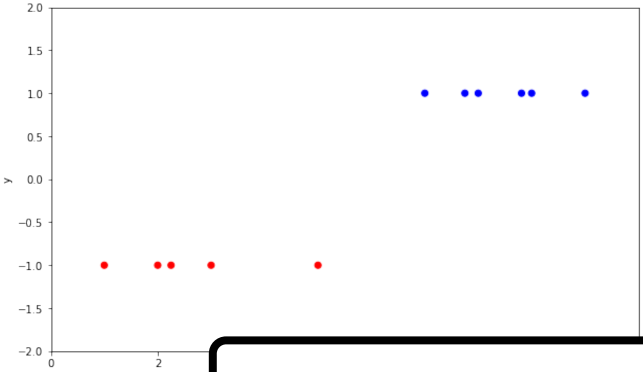
Binary classification using linear regression

Negatives: $y = -1$ Positives: $y = +1$



Binary classification using linear regression

Negatives: $y = -1$ Positives: $y = +1$

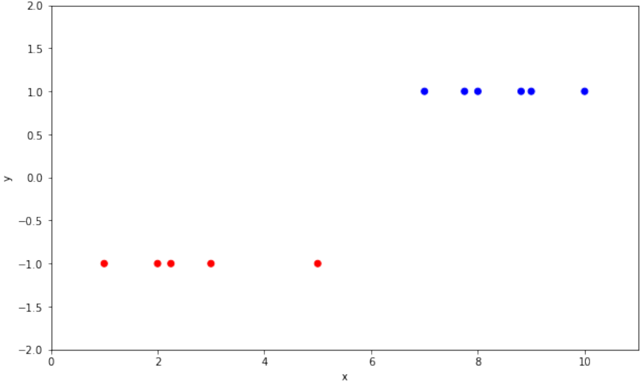


Determine $h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\,\mathbf{x}$ via linear regression

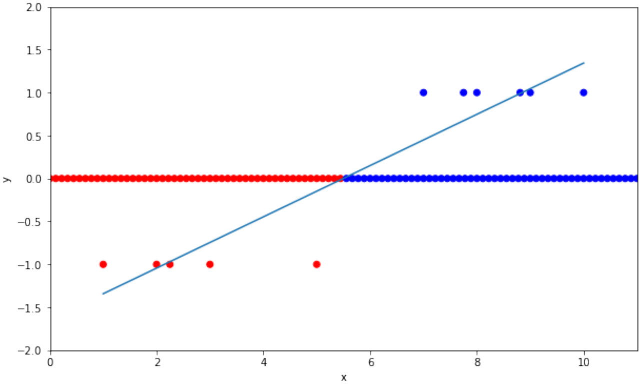
Classification: $\hat{y} = \mathrm{sign}\left(\mathbf{w}^T\,\mathbf{x}\right)$

Binary classification using linear regression

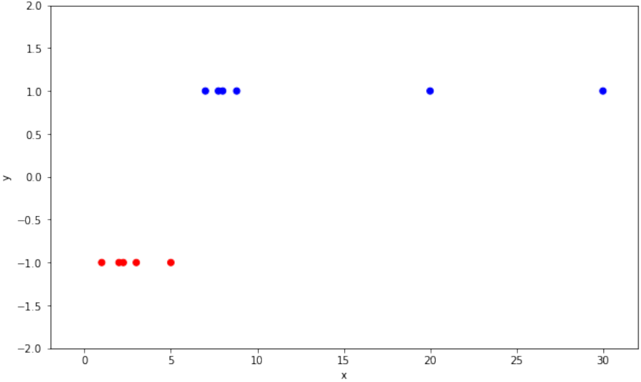
Negatives: $y = -1$ Positives: $y = +1$



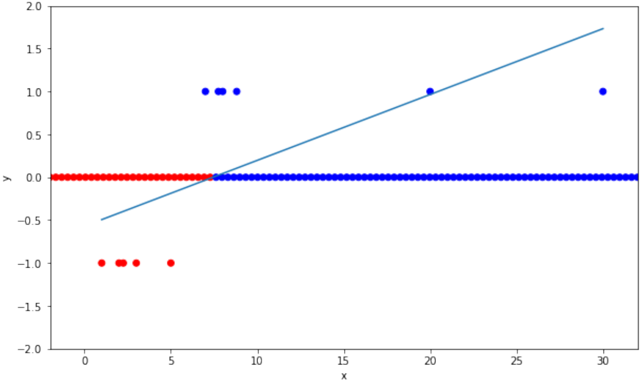
Binary classification using linear regression



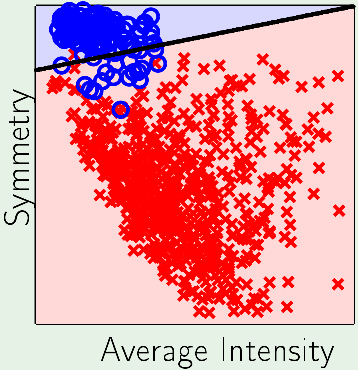
Second example: $D_2 = \{1, 2, 2.25, 3, 5, 7, 7.75, 8, 8.81, 20, 30\}$



Rightmost positive examples contribute with large error ...
Leftmost positive examples will be classified as negative ...



Linear regression boundary



Binary classification using linear regression

It somehow approximates the decision boundary

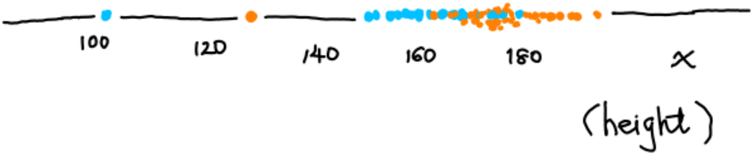
Strongly affected by how examples are scattered

There must be something better than it

Noisy targets

Examples usually are not perfectly separable

Two persons with a same height could be in distinct classes (Female and Male)



Noisy targets

The 'target function' is not always a *function*

Consider the credit-card approval:

age	23 years
annual salary	\$30,000
years in residence	1 year
years in job	1 year
current debt	\$15,000
$\cdots$	$\cdots$

two 'identical' customers $\longrightarrow$ two different behaviors

As we have discussed, from Bayes' Theorem we know that

$$P(y|\mathbf{x}) = \frac{P(y)p(\mathbf{x}|y)}{p(\mathbf{x})}$$

So, rather than trying to guess y, why not trying to estimate

$$P(y|\mathbf{x})$$

Instead of $y = f(\mathbf{x})$, our target would be the distribution

$$P(y|\mathbf{x})$$

Binary classification

Logistic Regression

Assume our target is: $f(\mathbf{x}) = P(y = +1|\mathbf{x})$

We can write

$$P(y|\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{if } y = +1, \\ 1 - f(\mathbf{x}), & \text{if } y = -1 \end{cases}$$

(we could have defined $f(\mathbf{x}) = P(y = -1|\mathbf{x})$)

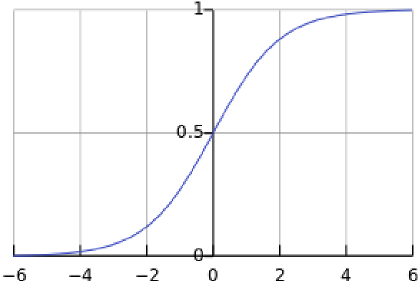
Note that we have no access to $f(\mathbf{x})$; we only know that y comes from a unknown distribution $P(y|\mathbf{x})$

But, if we are able to "learn" $f(\mathbf{x})$, we will be able to learn $P(y|\mathbf{x})$

Since our target f is such that $0 \leq f(\mathbf{x}) \leq 1$, let us consider hypotheses of the same type:

$$h_{\mathbf{w}}(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x})$$

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



$$0 \leq \theta(z) \leq 1 \quad \Longrightarrow \quad 0 \leq h_{\mathbf{w}}(\mathbf{x}) \leq 1$$

If $h_{\mathbf{w}}(\mathbf{x}) \approx f(\mathbf{x})$, then

$$P_{\mathbf{w}}(y|\mathbf{x}) = \begin{cases} h_{\mathbf{w}}(\mathbf{x}), & \text{if } y = +1, \\ 1 - h_{\mathbf{w}}(\mathbf{x}), & \text{if } y = -1 \end{cases}$$

should be a good estimate of $P(y|\mathbf{x})$

To avoid dealing separately with the two cases, $y = +1$ and $y = -1$,

note that $1 - \theta(z) = \theta(-z)$ (left as an exercise)

Using this fact plus $h_{\mathbf{w}}(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x})$, we can write

$$P_{\mathbf{w}}(y|\mathbf{x}) = \theta(y\,\mathbf{w}^T\mathbf{x})$$

Learning the target: $f(\mathbf{x}) = P(y = +1|\mathbf{x})$

Available training data:

$$\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)}) \in X \times Y, n = 1, \ldots, N\}$$

These examples follow an unknown joint distribution $P(\mathbf{x}, y)$.
y follows the distribution $P(y|\mathbf{x})$.

Among all distributions $P_{\mathbf{w}}(y|\mathbf{x}) = \theta(y\, \mathbf{w}^T\mathbf{x})$, which $\mathbf{w}$ best approximates $P(y|\mathbf{x})$?

Maximum likelihood estimation

We assume a parametric distribution, and find the parameters that correspond to the distribution that maximizes the likelihood of observing the actually observed examples

In our setting, among all distributions $P_{\mathbf{w}}(y|\mathbf{x}) = \theta(y\,\mathbf{w}^T\mathbf{x})$ (parameter $\mathbf{w}$) which is the one that maximizes the likelihood of observing the examples in D ?

Assuming examples in D are i.i.d., the **likelihood function** can be written as:

$$\prod_{n=1}^{N} P_{\mathbf{w}}(y^{(n)}|\mathbf{x}^{(n)}) = \prod_{n=1}^{N} \theta(y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)})$$

Optimization problem

Find $\mathbf{w}$ that <u>maximizes</u>

$$\prod_{n=1}^{N} \theta(y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)})$$

Or, equivalently, <u>maximizes</u>

$$\frac{1}{N}\ln\Big(\prod_{n=1}^{N} \theta(y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)})\Big)$$

Or, equivalently, <u>minimizes</u>

$$-\frac{1}{N}\ln\Big(\prod_{n=1}^{N} \theta(y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)})\Big)$$

We would like to find **w** that minimizes

$$-\frac{1}{N}\ln\left(\prod_{n=1}^{N}\theta\big(y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)}\big)\right)$$

$$-\frac{1}{N}\sum_{n=1}^{N}\ln\left(\theta\big(y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)}\big)\right) \quad \text{(since } \ln\prod a_i = \sum \ln a_i \text{)}$$

$$\frac{1}{N}\sum_{n=1}^{N}\ln\left(\frac{1}{\theta\big(y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)}\big)}\right) \quad \text{(since } \ln\tfrac{1}{a} = -\ln a \text{)}$$

$$\frac{1}{N}\sum_{n=1}^{N}\ln\left(1+e^{-y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)}}\right) \quad \text{(since } \tfrac{1}{\theta z} = \tfrac{1}{\frac{1}{1+e^{-z}}} \text{)}$$

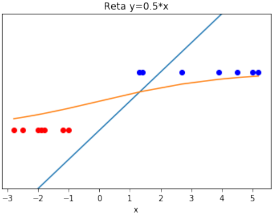
Logistic regression: Cost function to be minimized

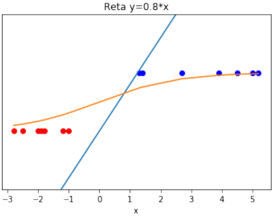
$$E_{in} = \frac{1}{N}\sum_{n=1}^{N} \underbrace{\ln\left(1 + e^{-y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)}}\right)}_{err(y^{(n)},\hat{y}^{(n)})}$$

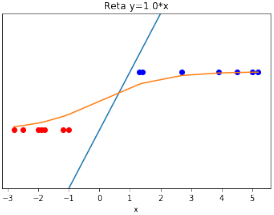
Interpretation:

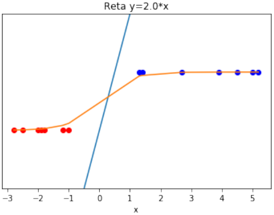
If the signals of $y^{(n)}$ and $\mathbf{w}^T\mathbf{x}^{(n)}$ agree, the exponent in $e^{-y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)}}$ is negative $\rightsquigarrow$ $err(y^{(n)}, \hat{y}^{(n)})$ tends to be close to zero

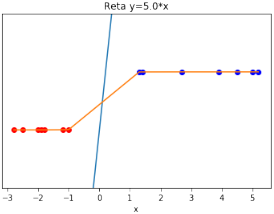
If the signals of $y^{(n)}$ and $\mathbf{w}^T\mathbf{x}^{(n)}$ disagree, the exponent in $e^{-y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)}}$ is positive $\rightsquigarrow$ $err(y^{(n)}, \hat{y}^{(n)})$ tends to be large











Blue line: $\mathbf{w}^T\,\mathbf{x} = w_0 + w_1\,x = 0$

Orange curve: $h_{\mathbf{w}}(\mathbf{x}) = \theta(\mathbf{w}^T\,\mathbf{x})$

Remarks:

$\Longrightarrow$ $y^{(n)} \in \{-1, +1\}$ while $\hat{y}^{(n)} \in [0, 1]$ ...

(we use $\hat{y}^{(n)}$ to indicate that this is the output of the algorithm, but it may not be the most adequate since $\hat{y}^{(n)} = \theta(\mathbf{w}^T\mathbf{x}) = P_{\mathbf{w}}(y = +1|\mathbf{x}^{(n)})$)

That is why the method is called logistic **regression**

$\Longrightarrow$ The formulation we have seen assumes $y \in \{-1, +1\}$
(this is also the formulation in the textbook)

$\Longrightarrow$ A more common (?) formulation assumes $y \in \{0, 1\}$

It is convenient that the logistic regression algorithm outputs $\hat{y}^{(n)} = \theta(\mathbf{w}^T\mathbf{x}) = P_{\mathbf{w}}(y = +1|\mathbf{x}^{(n)})$

Types of classification errors

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

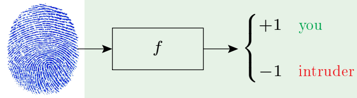
You can decide to classify $\mathbf{x}^{(n)}$ as positive only if $P_{\mathbf{w}}(y = +1|\mathbf{x}^{(n)}) \geq 0.8$. Conversely, $P_{\mathbf{w}}(y = +1|\mathbf{x}^{(n)}) \geq 0.3$ could make more sense in other cases.

The error measure - for supermarkets

Supermarket verifies fingerprint for discounts

False reject is costly; customer gets annoyed!

False accept is minor; gave away a discount and intruder left their fingerprint ☺



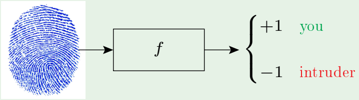
		f = $+1$	f = -1
h	$+1$	0	1
	-1	10	0

The error measure - for the CIA

CIA verifies fingerprint for security

False accept is a disaster!

False reject can be tolerated
Try again; you are an employee ☺



		f	
		$+1$	-1
h	$+1$	0	1000
	-1	1	0

We will also see later that **class imbalance** is an important issue when designing classifiers.

Example: Security system of a building

Suppose you need to design a face recognition based system to be used in an access controlled building. Only registered persons can enter the building.

It is known that an unauthorized person trying to enter the building is a very rare event.

Thus, if you design a system that authorizes access to every one, you have a system that is right 99.9% of the time. But the system is useless!

Formulation when we use $Y = \{0, 1\}$ rather than $Y = \{-1, +1\}$

A trick to write $P(y|\mathbf{x})$ as a single equation:

$$
\begin{aligned}
P(y|\mathbf{x}) &= P(y=1|\mathbf{x})^y \, P(y=0|\mathbf{x})^{1-y} \\
&= P(y=1|\mathbf{x})^y \, [1 - P(y=1|\mathbf{x})]^{1-y}
\end{aligned}
$$

Likelihood function (index (n) omitted for a cleaner notation)

$$
\begin{aligned}
\prod_{(\mathbf{x},y)\in D} P(y|\mathbf{x}) &= \prod_{(\mathbf{x},y)\in D} P(y=1|\mathbf{x})^y \, [1 - P(y=1|\mathbf{x})]^{1-y} \\
&\approx \prod_{(\mathbf{x},y)\in D} [\theta(\mathbf{w}^T\mathbf{x})]^y \, [1 - \theta(\mathbf{w}^T\mathbf{x})]^{1-y} \\
&= \prod_{(\mathbf{x},y)\in D} \hat{y}^y \, (1 - \hat{y})^{1-y}
\end{aligned}
$$

Likelihood function maximization:

$$\prod_{(\mathbf{x},y)\in D} \hat{y}^{y}\,(1-\hat{y})^{1-y}$$

Equivalent to minimization of:

$$-\ln \prod_{(\mathbf{x},y)\in D} \hat{y}^{y}\,(1-\hat{y})^{1-y}$$

$$-\sum_{(\mathbf{x},y)\in D} \ln\left(\hat{y}^{y}\,(1-\hat{y})^{1-y}\right)$$

$$-\sum_{(\mathbf{x},y)\in D} \ln(\hat{y}^{y}) + \ln((1-\hat{y})^{1-y})$$

$$-\sum_{(\mathbf{x},y)\in D} y\ln\hat{y} + (1-y)\ln(1-\hat{y})$$

Cross-entropy loss

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} y^{(n)} \ln \hat{y}^{(n)} + (1 - y^{(n)}) \ln(1 - \hat{y}^{(n)})$$

where $\hat{y}^{(n)} = \theta(\mathbf{w}^T\mathbf{x})$

Given two distributions p and q over A, cross-entropy is defined as:

$$H(p,q) = -\sum_{a\in A} p(a) \log q(a)$$

Cost functions

Textbook's formulation ($Y \in \{-1, +1\}$)

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y^{(n)}\mathbf{w}^T\mathbf{x}^{(n)}}\right)$$

Cross-entropy loss ($Y \in \{0, 1\}$, $\hat{y}^{(n)} = \theta(\mathbf{w}^T\mathbf{x}^{(n)})$)

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} y^{(n)} \ln \hat{y}^{(n)} + (1 - y^{(n)}) \ln(1 - \hat{y}^{(n)})$$

We use gradient descent to optimize them!

Optimization using gradient descent: Textbook's formulation

$$E_{in} = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + e^{-y^{(n)}\,\mathbf{w}^T\mathbf{x}^{(n)}}\right)$$

Gradient: $\frac{\partial}{\partial \mathbf{w}}[\ln\left(1 + e^{-y\,\mathbf{w}^T\mathbf{x}}\right)] = ?$

Denote $\mathbf{s} = -y\mathbf{x}$. Then $\frac{\partial}{\partial \mathbf{w}}[\ln\left(1 + e^{\mathbf{w}^T\mathbf{s}}\right)] = ?$

Since $\frac{\partial}{\partial \mathbf{w}}[\ln[f(x)] = \frac{f'(x)}{f(x)}$, then

$\frac{\partial}{\partial \mathbf{w}}[\ln(1 + e^{\mathbf{w}^T\mathbf{s}})] = \frac{(1+e^{\mathbf{w}^T\mathbf{s}})'}{1+e^{\mathbf{w}^T\mathbf{s}}} = \frac{\mathbf{s}\,e^{\mathbf{w}^T\mathbf{s}}}{1+e^{\mathbf{w}^T\mathbf{s}}} = \mathbf{s}\frac{e^{\mathbf{w}^T\mathbf{s}}}{1+e^{\mathbf{w}^T\mathbf{s}}} = \mathbf{s}\frac{1}{1+e^{-\mathbf{w}^T\mathbf{s}}}$

Hence,

$$\frac{\partial}{\partial \mathbf{w}}[\ln\left(1 + e^{-y\,\mathbf{w}^T\mathbf{x}}\right)] = -\frac{y\mathbf{x}}{1 + e^{y\mathbf{w}^T\mathbf{x}}}$$

Optimization using gradient descent: cross-entropy loss

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N} y^{(n)} \ln \hat{y}^{(n)} + (1 - y^{(n)}) \ln(1 - \hat{y}^{(n)})$$

$$\hat{y} = h_{\mathbf{w}}(\mathbf{x}) = \theta(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}}$$

... after some steps ...

Partial derivatives: $$\frac{\partial}{\partial w_j} J(\mathbf{w}) = \sum_{n=1}^{N} (\hat{y}^{(n)} - y^{(n)}) x_j^{(n)}$$

Weight update: $$\Delta w_j(r) = \sum_{n=1}^{N} (y^{(n)} - \hat{y}^{(n)}) \mathbf{x}_j^{(i)}$$

A third linear model

$$s = \sum_{i=0}^{d} w_i x_i$$

linear classification	linear regression	**logistic regression**
$h(\mathbf{x}) = \mathrm{sign}(s)$	$h(\mathbf{x}) = s$	$h(\mathbf{x}) = \theta(s)$